



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Dpto. INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

IMPLEMENTACIÓN DE LAS COMUNICACIONES PC-AUTÓMATA-ROBOT MEDIANTE INTERFAZ ETHERNET INDUSTRIAL

PROYECTO FIN DE CARRERA

INGENIERÍA TÉCNICA INDUSTRIAL: ELECTRÓNICA

AUTOR: MIGUEL ÁLVARO YUNTA

TUTOR: SANTIAGO MARTÍNEZ DE LA CASA

Agradecimientos.

En primer lugar agradezco a mis padres, y a mi familia en general, su apoyo durante todos estos años.

Agradezco también la ayuda proporcionada por el tutor del proyecto Santiago Martínez de la Casa, que siempre me ayudó a sacar este proyecto adelante. Y también agradezco a mis compañeros Dani y Sito, que me proporcionaron material para realizar este proyecto y con los cuales la carrera se hizo más amena.

Gracias a todos.



Índice.



| | |
|---|----------|
| 1. INTRODUCCIÓN | 1 |
| 1.0. Motivación del proyecto | 2 |
| 1.1. Objetivos | 2 |
| 1.2. Estructura de la memoria | 3 |
| | |
| 2. ESTADO DEL ARTE DE LAS COMUNICACIONES INDUSTRIALES. | |
| 2.0. Funciones de las redes de comunicación | 6 |
| 2.1. Niveles de comunicación en una red industrial | 7 |
| 2.2. Sistemas Distribuidos de Tiempo Real | 9 |
| 2.3. Ethernet Industrial | 10 |
| 2.4. Buses de campo | 14 |
| 2.4.1. Características y ámbitos de aplicación | 14 |
| 2.4.2. Arquitecturas de comunicación | 16 |
| 2.4.3. Buses de campo estandarizados | 19 |
| 2.5. El bus de campo PROFIBUS | 23 |
| 2.6. El bus de campo WORLDVIP | 25 |
| 2.7. El bus de comunicaciones CAN | 28 |
| 2.8. Buses y protocolos en Domótica e Inmótica | 30 |
| 2.9. Otros tipos de comunicación | 33 |
| 2.10. Situación actual y tendencias | 36 |



| | |
|---|-----------|
| 3. DESCRIPCIÓN DEL SISTEMA | 39 |
| 3.0. Condiciones de partida | 40 |
| 3.1. Descripción del hardware utilizado | 42 |
| 3.1.1. PC | 42 |
| 3.1.2. Autómata Siemens S7 314C | 43 |
| 3.1.3. Controlador IRC5 de robot ABB IRB 2400 | 45 |
| 3.1.4. Cable de red Ethernet RJ45 | 47 |
| 3.2. Descripción del software utilizado | 49 |
| 3.2.1. Step7 Simatic Manager | 49 |
| 3.2.2. Lenguaje RAPID para la programación del robot | 53 |
| 3.2.3. Wincc (Windows Control Center) | 54 |
| 4. DESARROLLO DEL PROYECTO | 57 |
| 4.1. Comunicación PC – autómata | 58 |
| 4.1.1. Objetivo, arquitectura y diagrama de flujo | 58 |
| 4.1.2. Configuración y programación en Wincc Creación de variables | 60 |
| 4.1.3. Creación de pantallas y uso variables en WinCC | 70 |
| 4.1.4. Configuración y programación en Simatic Manager Creación de la red Ethernet | 82 |



| | | |
|----------|---|-----|
| 4.2. | Comunicación autómatas – robot | 92 |
| 4.2.1. | Objetivo, arquitectura y diagrama de flujo | 92 |
| 4.2.2. | Conexión del robot a la red Ethernet. Creación de enlace TCP | 95 |
| 4.2.3. | Envío y recepción de datos | 102 |
| 4.2.3.1. | Autómata. Programación en Step7 | 102 |
| 4.2.3.2. | Robot. Programación RAPID. Sockets | 117 |
| 4.3. | Integración en el sistema PC-autómata-robot | 121 |
| 4.3.1. | Esquema de conexiones del sistema | 121 |
| 4.3.2. | Diagramas de flujo | 122 |
| 4.3.3. | Ejemplo de aplicación SCADA | 126 |
| 5. | CONCLUSIONES | 133 |
| 6. | TRABAJOS FUTUROS | 136 |
| 7. | BIBLIOGRAFÍA | 138 |



Índice de figuras.



CAPÍTULO 1: INTRODUCCIÓN

| | |
|--|---|
| Figura 1.1. Esquema de la aplicación desarrollada..... | 2 |
|--|---|

CAPÍTULO 2: ESTADO DEL ARTE DE LAS COMUNICACIONES INDUSTRIALES

| | |
|--|----|
| Figura 2.1. Jerarquía de comunicaciones industriales..... | 7 |
| Figura 2.2. La pila OSI..... | 8 |
| Figura 2.3. Ejemplo de arquitectura de un SCD..... | 10 |
| Figura 2.4. Ejemplo de arquitectura de una red Ethernet..... | 13 |
| Figura 2.5. Interacción cliente/servidor típica..... | 18 |
| Figura 2.6. Perfiles de protocolos PROFIBUS..... | 23 |
| Figura 2.7. Ejemplo de estructura física de una red WorldFIP..... | 26 |
| Figura 2.8. Esquema de conexiones; interfaz Industrial Ethernet y MPI..... | 34 |

CAPÍTULO 3: DESCRIPCIÓN DEL SISTEMA

| | |
|---|----|
| Figura 3.1. Diagrama de bloques del sistema..... | 40 |
| Figura 3.2. PC - SCADA..... | 42 |
| Figura 3.3. Puertos de comunicación del PC..... | 42 |
| Figura 3.4. Adaptador USB a MPI..... | 43 |
| Figura 3.5. Autómata Siemens S7 314C..... | 43 |
| Figura 3.6. Switch Industrial Ethernet SCALANCE X208..... | 44 |
| Figura 3.7. Controlador IRC5; Puerto Ethernet externo (servicio)..... | 46 |
| Figura 3.8. Controlador IRC5; Puerto Ethernet interno (red LAN)..... | 46 |
| Figura 3.9. Robot ABB IRB 2400..... | 47 |
| Figura 3.10. Dimensiones del robot..... | 47 |
| Figura 3.11. Cable Ethernet, conectores RJ45..... | 47 |
| Figura 3.12. Cable directo; Distribución 568A..... | 48 |
| Figura 3.13. Cable directo; Distribución 568B..... | 48 |
| Figura 3.14. FlexPendant del Controlador de robot IRC5..... | 53 |
| Figura 3.15. Estructura típica de una aplicación RAPID..... | 54 |



CAPÍTULO 4: DESARROLLO DEL PROYECTO

| | |
|---|----|
| Figura 4.1. Comunicación PC – Autómata..... | 58 |
| Figura 4.2. Arquitectura de comunicación PC – Autómata..... | 59 |
| Figura 4.3. Diagrama de flujo de la comunicación PC – Autómata..... | 59 |
| Figura 4.4. Creación de un proyecto monopuesto en WinCC..... | 60 |
| Figura 4.5. Administrador de variables en WinCC..... | 61 |
| Figura 4.6. Agregar driver de comunicación de WinCC..... | 61 |
| Figura 4.7. Interfaces de comunicación del driver Simatic S7 Protocol Suite..... | 62 |
| Figura 4.8. Creación de una nueva conexión por Industrial Ethernet..... | 62 |
| Figura 4.9. Propiedades de la nueva conexión Ethernet creada..... | 63 |
| Figura 4.10. Parámetros de la conexión por Industrial Ethernet..... | 63 |
| Figura 4.11. Creación de variables en Industrial Ethernet..... | 64 |
| Figura 4.12. Establecimiento de las propiedades de la variable Ethernet creada..... | 65 |
| Figura 4.13. Direccionamiento de la variable..... | 66 |
| Figura 4.14. Variables utilizadas en la comunicación por Industrial Ethernet..... | 67 |
| Figura 4.15. Creación de una nueva conexión MPI..... | 68 |
| Figura 4.16. Propiedades de la conexión MPI..... | 68 |
| Figura 4.17. Parametrización del enlace MPI..... | 69 |
| Figura 4.18. Variables de la conexión MPI..... | 69 |
| Figura 4.19. Propiedades de la variable MPI..... | 70 |
| Figura 4.20. Abrir el editor de pantallas Graphics Designer..... | 71 |
| Figura 4.21. Pantalla de inicio del SCADA..... | 72 |
| Figura 4.22. Paleta de objetos del Graphics Designer..... | 72 |
| Figura 4.23. Configuración del botón ACCEDER..... | 73 |
| Figura 4.24. Asistente para salir de Runtime..... | 73 |
| Figura 4.25. Ajuste de programas ejecutados al iniciar la Runtime..... | 74 |
| Figura 4.26. Pantalla de selección de operación..... | 74 |
| Figura 4.27. Asistente dinámico para activar/desactivar bit..... | 75 |



| | |
|---|----|
| Figura 4.28. Selección del momento de activación del evento..... | 75 |
| Figura 4.29. Selección de la acción a ejecutar tras evento..... | 76 |
| Figura 4.30. Selección del bit..... | 76 |
| Figura 4.31. Pantalla de diagnóstico de las comunicaciones..... | 77 |
| Figura 4.32. Selección del objeto Vista de estado..... | 77 |
| Figura 4.33. Configuración de imágenes del objeto Vista de estado..... | 78 |
| Figura 4.34. Visualización del estado de las comunicaciones..... | 78 |
| Figura 4.35. Pantalla de manejo de variables Ethernet..... | 79 |
| Figura 4.36. Bloque de recepción de datos en el autómatas..... | 80 |
| Figura 4.37. Animación de imágenes de la tarea ejecutada por el robot..... | 80 |
| Figura 4.38. Configuración de textos descriptivos de la tarea ejecutada por el robot...81 | |
| Figura 4.39. Bloque para el envío de datos desde el autómatas..... | 81 |
| Figura 4.40. Selección de la tarjeta Ethernet del PC como interfaz..... | 82 |
| Figura 4.41. Insertar el autómatas SIMATIC 300..... | 83 |
| Figura 4.42. Configuración hardware del autómatas..... | 83 |
| Figura 4.43. Insertar bastidor del autómatas 1/2..... | 83 |
| Figura 4.44. Insertar bastidor del autómatas 2/2..... | 84 |
| Figura 4.45. Configuración hardware del autómatas, Fuente de Alimentación..... | 84 |
| Figura 4.46. Configuración hardware del autómatas, CPU..... | 85 |
| Figura 4.47. Configuración hardware del autómatas, Procesador de Comunicaciones...86 | |
| Figura 4.48. Editar estación Ethernet..... | 87 |
| Figura 4.49. Buscar estaciones Ethernet online..... | 87 |
| Figura 4.50. Estaciones Ethernet..... | 88 |
| Figura 4.51. Insertar un PC en la red..... | 88 |
| Figura 4.52. Conjunto de elementos insertados en la red..... | 89 |
| Figura 4.53. Asociar PC a red Ethernet..... | 89 |
| Figura 4.54. Conexión Ethernet PC..... | 89 |
| Figura 4.55. Estado conexión Ethernet PC..... | 90 |
| Figura 4.56. Protocolo TCP/IP Ethernet..... | 90 |
| Figura 4.57. Propiedades del protocolo TCP/IP en el PC..... | 91 |
| Figura 4.58. Red de comunicación entre el PC y el autómatas..... | 91 |
| Figura 4.59. Comunicación Autómatas – Robot..... | 92 |
| Figura 4.60. Arquitectura de comunicación Autómatas – Robot..... | 93 |
| Figura 4.61. Diagrama de flujo de la comunicación Autómatas – Robot..... | 94 |



| | |
|--|-----|
| Figura 4.62. Insertar equipo ajeno (IRC5)..... | 95 |
| Figura 4.63. Vista de los elementos que componen la red..... | 95 |
| Figura 4.64. Creación de interfaz Industrial Ethernet para el IRC5..... | 96 |
| Figura 4.65. Asociación del IRC5 a la red Ethernet..... | 96 |
| Figura 4.66. Abrir herramienta NetPro..... | 98 |
| Figura 4.67. Creación de enlace TCP para comunicación con autómatas..... | 99 |
| Figura 4.68. Selección del tipo de enlace..... | 99 |
| Figura 4.69. Propiedades del enlace TCP..... | 100 |
| Figura 4.70. Direcciones de enlace TCP..... | 100 |
| Figura 4.71. Selección de interfaz SEND/RECEIVE para el enlace TCP..... | 101 |
| Figura 4.72. Herramienta de configuración de red Netpro..... | 102 |
| Figura 4.73. Guardar y compilar la configuración del Netpro..... | 102 |
| Figura 4.74. Cambio del lenguaje de programación..... | 103 |
| Figura 4.75. Estructura del programa Step7..... | 103 |
| Figura 4.76. Bloques del programa implementado en el S7..... | 106 |
| Figura 4.77. Bloques de la librería SIMATIC NET CP..... | 106 |
| Figura 4.78. Abrir configuración hardware del autómatas..... | 107 |
| Figura 4.79. Configuración del byte 10 como marca de ciclo..... | 108 |
| Figura 4.80. Guardar, compilar y cargar la configuración hardware en el PLC..... | 108 |
| Figura 4.81. Propiedades del enlace TCP en el Netpro..... | 109 |
| Figura 4.82. Obtención de parámetros ID y LADDR del enlace TCP..... | 109 |
| Figura 4.83. Tabla de variables creada..... | 116 |
| Figura 4.84. Buffer de envío y recepción; longitud 1 byte..... | 116 |
| Figura 4.85. Valores de estado y forzado de los datos comunicados..... | 116 |
| Figura 4.86. Recepción de datos en el FlexPendant del IRC5..... | 117 |
| Figura 4.87. Diagrama de flujo del programa RAPID..... | 118 |
| Figura 4.88. Esquema de bloques e interfaces de comunicación del sistema..... | 121 |
| Figura 4.89. Esquema de conexiones; interfaz Ethernet Industrial..... | 122 |
| Figura 4.90. Diagrama de flujo; Envío de datos PC → Controlador de Robot..... | 123 |
| Figura 4.91. Diagrama de flujo; Envío de datos Controlador de Robot → PC..... | 124 |
| Figura 4.92. Pantalla 1 del SCADA ejemplo..... | 127 |
| Figura 4.93. Pantalla 2 del SCADA ejemplo..... | 128 |
| Figura 4.94. Pantalla 3 del SCADA ejemplo..... | 129 |
| Figura 4.95. Pantalla 4 del SCADA ejemplo..... | 130 |



Índice de tablas.



CAPÍTULO 2: ESTADO DEL ARTE DE LAS COMUNICACIONES INDUSTRIALES

| | |
|--|----|
| Tabla 2.1. Modelos de Sistemas Distribuidos..... | 17 |
| Tabla 2.2. Relación estándares buses de campo CENELEC e IEC..... | 19 |
| Tabla 2.3. Comparativa de los principales buses de campo..... | 20 |
| Tabla 2.4. Medios físicos de las redes domóticas..... | 30 |

CAPÍTULO 3: DESCRIPCIÓN DEL SISTEMA

| | |
|---|----|
| Tabla 3.1. Parámetros característicos del controlador IRC5..... | 45 |
|---|----|

CAPÍTULO 4: DESARROLLO DEL PROYECTO

| | |
|---|-----|
| Tabla 4.1. Variables de entrada y salida..... | 125 |
| Tabla 4.2. Direcciones IP de los dispositivos de red..... | 125 |
| Tabla 4.3. Vistas de estado del proceso 1/2..... | 131 |
| Tabla 4.4. Vistas de estado del proceso 2/2..... | 132 |



Capítulo 1:

Introducción.

1.0. Motivación del proyecto.

Toda máquina debe ser controlada por un ser humano. Para ello es necesario el intercambio de información entre la máquina y las personas. Esta transferencia de información es posible a través de las redes de comunicación, siendo la implementación de éstas el motivo principal que nos llevó a la realización del presente proyecto.

En concreto, nos comunicaremos con un robot ABB mediante la red de comunicación Ethernet Industrial empleando un PC y un autómata. El uso de Ethernet Industrial se debe, aparte de las ventajas que aporta como son bajo coste y rapidez, a que es una red que no había sido implementada previamente en el laboratorio del Departamento de Ingeniería de Sistemas y Automática, ya que su uso en el entorno industrial todavía no está muy extendido, aunque cada vez es mayor.

1.1. Objetivos del proyecto.

El objetivo principal de este proyecto es la implementación de una red de comunicación Ethernet formada por un PC, un autómata Siemens S7 y el controlador de robot ABB IRC5.

La aplicación desarrollada deberá permitirnos una comunicación en dos sentidos, es decir, tanto desde el PC como desde el controlador del robot podrán enviarse y recibirse datos. En el esquema siguiente se muestra esta funcionalidad:

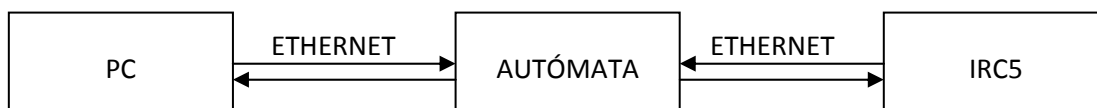


Figura 1.1. Esquema de la aplicación desarrollada.

Como objetivos secundarios podrían nombrarse los siguientes:

- Implementar la comunicación PC – Autómata. Desde el ordenador, mediante el software WinCC, podremos consultar el estado de posiciones de memoria del autómata (lectura de datos) así como forzar valores en éste (escritura de datos).
- Implementar la comunicación Autómata – Controlador de Robot. Se desarrollarán programas para hacer posible el envío y la recepción de datos desde ambos elementos. En el controlador de robot usaremos el lenguaje RAPID y emplearemos *sockets* como canal de comunicación, y el autómata lo programaremos mediante el software de Siemens SIMATIC Manager.
- Proporcionar al proyecto Manubuild la base para la creación de un sistema SCADA (Supervisory Control and Data Acquisition). Desde la pantalla del PC podremos visualizar qué tarea realiza el robot en todo momento (recepción de datos), así como enviar información al IRC5, en este caso lo que se envía es un byte.
- Crear una documentación que sirva de base para desarrollar prácticas de asignaturas en las que se requiere comunicación externa con los robots ABB IRB 1600 y ABB IRB 2400.

1.2. Estructura de la memoria.

El tema fundamental en torno al cual gira el presente proyecto es el de las comunicaciones industriales. Por ello, en el **Capítulo 2** se exponen conceptos básicos muy ligados a una red de comunicación, y se realiza un análisis de la situación actual de los sistemas de comunicación en la industria, explicando brevemente las características de las redes más utilizadas.

En el **Capítulo 3** se detallan las especificaciones del hardware y del software con el que se ha trabajado para la realización del proyecto.



En el **Capítulo 4** se muestra paso a paso el desarrollo del proyecto. En primer lugar nos centraremos en la comunicación PC – Autómata, luego pasaremos a la Comunicación Autómata – Robot, y por último se tratará la integración de los tres elementos que constituyen el sistema de comunicación implementado.

El **Capítulo 5** está dedicado a las conclusiones. En él se analiza si se han cumplido los objetivos inicialmente propuestos, y se expone una conclusión sobre la interfaz de comunicación empleada (Ethernet Industrial). También en este punto se habla sobre las dificultades y problemas surgidos durante la realización del proyecto, y se indica la solución de los mismos.

En el **Capítulo 6** se habla sobre los trabajos futuros que podrían llevarse a cabo sobre la base de lo explicado en el presente documento.



Capítulo 2:
Estado del Arte de las Comunicaciones Industriales.



Las comunicaciones industriales son una de las áreas en auge dentro del amplio mundo de las comunicaciones, al unirse en un mismo entorno los temas empresariales (y más en detalle, los temas de fabricación) y las comunicaciones, como soporte para la implantación tecnológica dentro de la empresa.

2.0. Funciones de las redes de comunicación.

Los objetivos de todo sistema de comunicación en el entorno industrial son:

- Coordinar acciones de unidades automatizadas y controlar la transferencia de componentes, a través del intercambio de datos entre las diferentes unidades (autómatas programables y PCs industriales) que controlan el proceso productivo.
- Monitorizar y modificar estrategias de control desde el puesto de operación, que puede estar situado en la propia planta o en cualquier otro lugar mediante una conexión a través de redes de datos públicas o privadas.
- Proveer los recursos necesarios para aumentar la confiabilidad y seguridad en los procesos de producción mediante: detección temprana de condiciones de alarma, supervisión y control continuo de procesos de alto riesgo, verificación del estado de las instalaciones y seguimiento de las condiciones de operación de estaciones remotas.
- Proveer servicios de transmisión de voz e imágenes.
- Integración completa del proceso productivo (desde el operario hasta los gestores o clientes).

Como se puede observar, desde el punto de vista industrial, la necesidad de comunicación no se restringe únicamente a la producción. Diferentes departamentos de la industria pueden participar en la red de comunicaciones para permitir un control global del sistema. De este modo, no sólo se controla el propio funcionamiento de la planta de fabricación, sino que en función de las decisiones tomadas en las capas administrativas de la empresa, podría actuarse directamente sobre la producción.

Por lo tanto, la red integrada de comunicación industrial debe estructurarse en base a una arquitectura bien definida y bajo las premisas de racionalización, conectividad, calidad y confianza.

2.1. Niveles de comunicación en una red industrial.

La integración de los diferentes equipos y dispositivos existentes en una industria se hace dividiendo las tareas entre grupos de procesadores con una organización jerárquica. Así, dependiendo de la función y el tipo de conexiones, se suelen distinguir cinco niveles en una red industrial:

- N1: Nivel de entrada/salida.
- N2: Nivel de campo.
- N3: Nivel de control de proceso.
- N4: Nivel de control de producción.
- N5: Nivel de gestión o dirección.

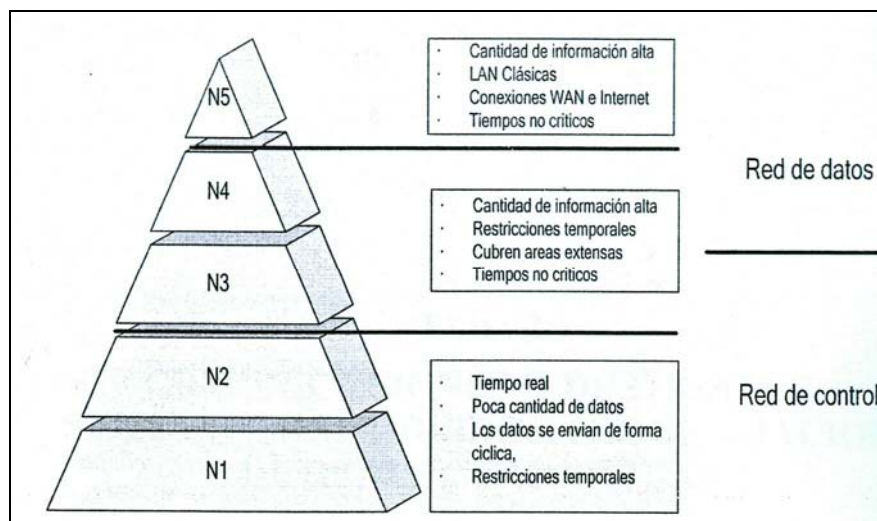


Figura 2.1. Jerarquía de comunicaciones industriales.

Esta estructura no es universal, varía con el tamaño del sistema de fabricación y sus características particulares. Además, para cualquiera de los niveles, no hay un estándar universalmente aceptado que cubra todos los aspectos desde el nivel físico al de aplicación (si nos referimos al modelo OSI de ISO).



Figura 2.2. La pila OSI.

Este modelo OSI de 7 capas propuesto en 1984 se mostró ineficiente para su utilización en redes industriales con requerimientos de baja latencia, debido a la sobrecarga que este modelo impone en cada capa. Para solventar este inconveniente, la mayoría de redes industriales utilizaron únicamente 3 de estos niveles, el nivel físico, el nivel de enlace y el nivel de aplicación, por lo que a continuación se repasan las particularidades de estos niveles en el entorno industrial.

El nivel físico define el medio físico que se utiliza en la transmisión y las características físicas del mismo, como niveles de voltaje, sistema de codificación, etc. Estas características determinan la topología, la velocidad de transmisión, el número máximo de nodos en una red, etc.

El nivel de enlace define los formatos de trama, mecanismos de protección ante errores en la transmisión (CRC o código de redundancia cíclica). En la mayoría de



redes, incluyendo los buses de campo, en este nivel se ubica el subnivel de acceso al medio (MAC, *Medium Access Control*). La capacidad de satisfacer los requerimientos de tiempo real de las aplicaciones industriales dependerá en gran medida de si el mecanismo de acceso al medio tiene un comportamiento determinista o no determinista.

El nivel de aplicación define los interfaces entre el usuario y el sistema, y habitualmente incluye el nivel de usuario, denominado así dado que habitualmente es la forma en que el usuario ve el bus de campo, aislándole del manejo de los niveles inferiores. Los estándares proponen a este nivel objetos específicos para diferentes dominios de aplicación (robótica, control numérico, control de procesos, etc.).

2.2. Comunicaciones Industriales en Sistemas Distribuidos de Tiempo Real.

Las redes de comunicaciones industriales están muy ligadas a los sistemas de tiempo real (sistemas que interaccionan repetidamente con su entorno físico y responden a los estímulos que reciben del mismo dentro de un plazo determinado). En estos sistemas para que el funcionamiento sea correcto no basta con que las acciones sean correctas, sino que tiene que ejecutarse dentro de un intervalo de tiempo específico. El conjunto de todos los nodos interconectados en la red, junto con los servicios y protocolos usados para permitir el intercambio correcto y a tiempo de los datos, conforman lo que se conoce como sistema de comunicación de tiempo real.

En el caso de un sistema distribuido¹ de control de tiempo real si las salidas correctas no son producidas a tiempo, podrían producirse excesivos retardos de procesamiento y de comunicación, y el proceso bajo control podría hacerse inestable y poner en peligro toda la integridad del sistema. Cuando en el sistema distribuido existen nodos que ejecutan tareas de control se tiene un sistema distribuido de control (SCD). En este

¹ Los sistemas distribuidos están formados por dispositivos autónomos inteligentes que cooperan con objetivos concretos y que ejecutan tareas que se coordinan entre sí intercambiando información por medio de una red de comunicación.

caso, la red de comunicación se denomina bus de campo. Las generalidades de este tipo de redes son explicadas más adelante en el punto 2.5.

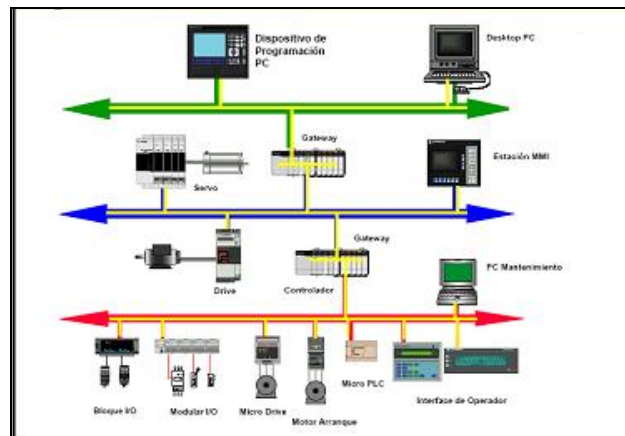


Figura 2.3. Ejemplo de arquitectura de un SCD.

2.3. Ethernet Industrial.

Cada vez más crece la aceptación que Ethernet tiene en el campo industrial. Esto es debido fundamentalmente a las ventajas que aporta cuando es instalado en la industria (rapidez, simple ampliación y apertura y costes de instalación y configuración muy bajos entre otras), a la forma de interconectarlo y a las posibilidades de diagnosticar su funcionamiento. Además otra característica importante que presenta es que permite la comunicación por protocolo TCP/IP con equipos ajenos (como por ejemplo el controlador de robot IRC5).

En pocos años ha podido verse un cambio interesante en las redes industriales. Adoptar Ethernet en el mundo de la automatización ha sido el principio de este cambio. La primera razón de su uso es sin duda que los usuarios potenciales de automatización ven a priori tener una base de conocimientos más elevada en redes basadas en Ethernet que no en soluciones de bus de campo propietarias y convencionales.



Actualmente Industrial Ethernet aún no domina el mercado de la comunicación industrial ya que tienen algunas deficiencias como es que los segmentos de cable se limitan a 100 metros, necesita concentradores, y los concentradores requieren un puerto cada uno para poder conectarse. Por otra parte Ethernet provee a los sistemas de control de más ancho de banda y puede tener redundancia de medio físico. Podemos decir que las principales razones de su uso son las siguientes:

- Interoperabilidad, que es la posibilidad de conectar dos dispositivos a través del mismo medio sin conflictos. Los dispositivos deben poder abordar el mismo medio físico. Esto da a las soluciones basadas en Ethernet la posibilidad de que varios protocolos puedan convivir entre ellos, utilizando si es necesario el mismo medio físico que se usa para conectar los PCs de las oficinas, las cámaras de vigilancia, e incluso las impresoras de red.
- Plug'n'Play: mediante este sistema se facilita el trabajo al usuario y se reducen costes en el mantenimiento o instalaciones tediosas y largas. Esto se traduce en un sistema que minimice o incluso no necesite un manual de configuración o parametrización. La conectividad de Ethernet, y la facilidad de los equipos que utilizan esta tecnología, se encuentra al alcance de todos los usuarios, pudiendo realizar cualquier trabajador las labores de instalación y configuración mediante software de manera rápida y fácil.
- Robustez. Una caída de la red industrial puede ser extremadamente dañina a los servicios de control, así Ethernet está preparada para estos fallos. Ethernet suele utilizar varios niveles de redundancia que refuerzan las comunicaciones de los componentes. Adicionalmente a esta redundancia de cableado, Ethernet dispone de mecanismos que se utilizan para encontrar rutas alternativas a la comunicación cuando se produce un fallo en un posible camino.

A nivel físico, la fibra óptica utilizada en Ethernet puede utilizarse en varias áreas donde se necesiten cubrir grandes distancias e incluso en zonas donde la interferencia electromagnética sea muy elevada. Los conectores creados para



Industrial Ethernet soportan golpes, vibraciones y temperaturas extremas. El cableado por último también está más que preparado, ya que el par trenzado empleado posee una buena relación S/N (señal-ruido) y realiza un filtrado del ruido a altas frecuencias de transmisión.

- Software Ethernet: no se requiere instrumentación o software específico, para el control y gestión de la red pueden utilizarse todas las herramientas que existen actualmente. Existen varias herramientas muy útiles basadas en IP como el Ping que ayuda a ver si una de las estaciones ha caído, que pueden utilizarse en cualquier sistema basado en Ethernet con TCP/IP.
- OPC (*OLE for Process Control*). Se trata de un estándar de comunicación en el campo de control y supervisión de procesos. Permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con *drivers* para dialogar con múltiples fuentes de datos, basta que tengan un *driver* OPC.
- Control y mantenimiento remoto. A través de Ethernet y gracias a la tecnología IP, los controladores de un proceso pueden estar dotados con servidores Webs, y éstos tienen acceso a los datos del proceso. Así, podríamos fácilmente por ejemplo comprobar el estado de cientos de sensores, por los que se podría navegar si sus procesos estuvieran en un servidor Web. La reducción de costes por establecimiento de conexiones vía MODEM, impedir que la distancia sea una limitación al trabajo, y permitir la carga de programas vía Internet, son algunas de sus cualidades.
- Posibilidad de atravesar la frontera del cableado físico mediante la solución Wireless Ethernet.

A nivel físico, los *Hubs*, *Bridges*, *Routers* y *Switches* son el pilar básico de una red basada en Ethernet. Casi igual de importante es elegir estos dispositivos como el cableado que los une. Ethernet originalmente fue diseñada para trabajar con cableado coaxial, aunque ya prácticamente obsoleto suele utilizarse en su defecto el par trenzado UTP y las fibras ópticas. UTP se encuentra disponible en varias categorías. Utilizando las categorías más altas puede llegarse a soportar frecuencias de 100 Mhz. La fibra óptica ofrece una muy buena inmunidad al ruido y la posibilidad de cubrir distancias de cientos de metros en un único segmento. Las LAN (*Local Area Network*) tradicionales se diseñan conectando un área de trabajo con un simple *bridge* a una red central o *Backbone*. A continuación se muestra un ejemplo de arquitectura de red:

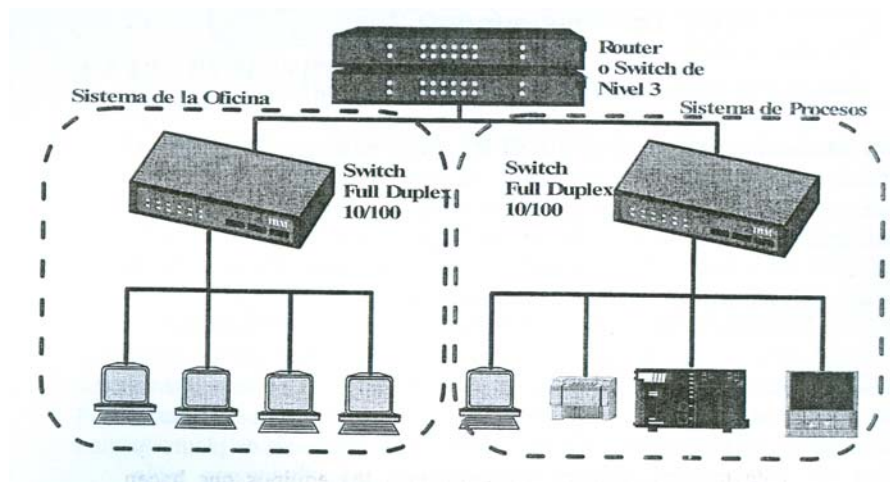


Figura 2.4. Ejemplo de arquitectura de una red Ethernet.

Los equipos de automatización se conectan mediante switches 10/100 Mbits/s *full duplex* como si fueran PCs tradicionales. Los equipos de planta, y los de oficina, se conectan a través de un switch con capacidad de nivel 3 (capacidad de leer paquetes a nivel de red), lo cual permite, si se requiere, aislar el tráfico de la oficina con el tráfico de la planta industrial. Este hecho provee a la red de un cierto grado de seguridad.

Actualmente existen unos 14 estándares diferentes en lo relativo a protocolos basados en Industrial Ethernet, todos ellos con sus características propias. Los principales son: EtherCAT, Ethernet PowerLink, Ethernet/IP y Profinet.



2.4. Buses de campo.

Según la definición elaborada por la institución Fieldbus Foundation, un bus de campo es un enlace de comunicaciones digital, bidireccional y multipunto entre dispositivos inteligentes de control y medida. Actúa como una red de área local para control de proceso avanzado, entrada/salida remota, y aplicaciones de automatización de alta velocidad. Los buses de campo son utilizados hoy en día en todo tipo de procesos de automatización que van desde la automatización industrial a la domótica, construcción de maquinaria, aplicaciones en sectores de automoción tanto automovilístico, como en ferrocarriles y aeronáutica.

2.4.1. Características y ámbitos de aplicación.

Una característica común de todos estos diferentes ámbitos es, como se ha comentado anteriormente, la necesidad de simplificar el cableado. Antes de la aparición de esta tecnología, los sensores analógicos transmitían su información con el estándar analógico de corriente 4-20 mA, y los digitales (verdadero, falso) mediante señales 0-24 V. El diseño, instalación, mantenimiento y reconfiguración de estos sistemas era complejo y costoso, por lo que la introducción de las comunicaciones digitales, y en particular los buses de campo basados en estas nuevas tecnologías, permitían simplificar el problema y reducir los costes.

Esta reducción de costes se alcanza no únicamente a través de la reducción de cableado, se estima que en una proporción de 5 a 1, sino que también influyen indirectamente otros de los requerimientos exigidos a esta tecnología. Uno de estos requerimientos importantes es la flexibilidad y la capacidad de añadir nuevos dispositivos a la red. Así, los sistemas basados en redes son mucho más fáciles de adaptar y evolucionar ante requerimientos de cambios, por ejemplo ampliando el número de dispositivos, que los sistemas centralizados tradicionales. Otra característica importante que deben cumplir es facilitar las tareas de mantenimiento. La red permite la monitorización de todos los elementos conectados, la actualización de software, y el diagnóstico, facilitando así tanto la puesta en marcha como el



mantenimiento de los sistemas. Otra propiedad es la capacidad de disponer de un canal bidireccional de comunicación con los dispositivos de campo, así como la capacidad de ofrecer un acceso remoto a la información de la red. A estas redes también se les exige una gran fiabilidad. Finalmente, esta tecnología permite implementar estrategias de control más avanzadas.

Estas redes, además, han de poder operar en entornos hostiles, es decir, entornos donde puede haber fuertes vibraciones, radiaciones electromagnéticas, ruido eléctrico, o en una atmósfera agresiva, lo que condiciona fuertemente los medios del nivel físico.

Vistas las características y propiedades de los buses de campo, de forma general, hay que recordar que esta tecnología se aplica en ámbitos con particularidades muy diferentes.

En el campo de la automatización industrial (sistemas discretos), esta tecnología cubre tanto la comunicación en una máquina concreta, como la comunicación entre máquinas. Los criterios de disponibilidad y fiabilidad son en este tipo de aplicaciones menos estrictos que en otros casos, como se verá a continuación.

En el campo del control de procesos (procesos continuos), las redes de comunicación deben cumplir unos requerimientos temporales estrictos de sincronización. En industrias como la química, se requiere seguridad intrínseca con la alimentación de dispositivos a través de la propia red. En algunas aplicaciones los elevados requerimientos de fiabilidad y seguridad requieren de redundancia para poder satisfacerlos.

En la domótica, las aplicaciones se consideran generalmente más del tipo de adquisición de datos que de control, o bien con funciones de control muy simples, por lo que es el ámbito de aplicación donde los requerimientos temporales son menos estrictos. Las particularidades más significativas de este tipo de aplicaciones son la gran variedad de sensores y dispositivos, así como el elevado número de éstos, lo que los convierte en un sistema complejo. El número de dispositivos y la necesidad de



reducir costes de cableado están disparando en este sector el uso de redes inalámbricas o de redes que usan los cables de tensión para la transmisión (*power line communications*). La fiabilidad, aún siendo importante, es menos estricta que en los casos anteriores.

En el campo de las redes de control para servicios públicos, las aplicaciones se centran en la monitorización y control de grandes redes de distribución (gas, agua, electricidad). En este entorno no se puede hablar de redes de área local (LAN), sino de redes metropolitanas (MAN) o de área extensa (WAN). Aunque pueden tener unos requerimientos de sincronización y fiabilidad elevados, la particularidad aquí es la distancia, a lo que se debe adaptar el medio físico.

En el campo de los sistemas de transporte, como por ejemplo el control de tráfico o la monitorización de autopistas y líneas de tren la topología de la red dependerá del área que se cubra. La seguridad y fiabilidad son aquí aspectos cruciales, puesto que vidas humanas pueden depender de ello.

En sistemas embebidos, pero especialmente en los usados en vehículos como coches, trenes, etc. Las distancias son pequeñas o muy pequeñas. Las funciones en este campo son muy variadas, desde el control de motores y frenos, a la gestión de luces y accesos, e incluso el pilotaje completo del vehículo. Las restricciones temporales serán dependientes de la aplicación en concreto, pero seguridad y fiabilidad son aquí otra vez aspectos cruciales.

2.4.2. Arquitecturas de comunicación.

Existen tres procedimientos o modelos distintos usados dentro de la comunicación en el área de las comunicaciones industriales:

- Modelo cliente-servidor.
- Modelo productor-consumidor.
- Modelo de publicación-suscripción.



| | Modelo cliente – servidor | Modelo productor – consumidor | Modelo de publicación – suscripción |
|-----------------------------|---|---|---|
| Tipo de comunicación | Entre iguales (<i>peer-to-peer</i>) | Difusión (<i>Broadcast</i>) | Multidifusión (<i>Multicast</i>) |
| Estilo de comunicación | Orientada a la conexión | Sin conexión explícita | Sin conexión explícita |
| Relación maestro-esclavo | Uno o varios maestros | Varios maestros | Varios maestros |
| Servicios de comunicaciones | Confirmados, sin confirmar, con confirmación | Sin confirmar, con confirmación | Sin confirmar, con confirmación |
| Clases de aplicaciones | Transferencia de parámetros, comunicación cíclica | Notificación de eventos, alarmas, eventos, sincronización | Cambios de estado y notificación de eventos |

Tabla 2.1. Modelos de Sistemas Distribuidos.

En la aplicación desarrollada en el presente proyecto se ha empleado una arquitectura punto a punto, donde el autómatas actúa como servidor y el robot como cliente. Dicho modelo de comunicación es explicado en el punto 4. Los procesos cliente y servidor se comunican a través del protocolo TCP/IP y la interfaz SOCKETS.

El paradigma de comunicación más ampliamente utilizado es el cliente/servidor. A través de este modelo, los procesos interactúan a través de peticiones y respuestas. El cliente es el proceso que hace las peticiones a otro proceso de la red, el servidor. El servidor realizará el trabajo, y devolverá al cliente un mensaje con los resultados. Es pues un modelo punto a punto. Este modelo se considera habitualmente más útil para la transmisión de información de estado, que para la comunicación de eventos. El servidor sólo podrá transmitir un evento tras una petición del cliente.

Las interacciones cliente/servidor se dan habitualmente en 4 pasos, como se ve en la siguiente figura. *Indication* es un evento producido en el servidor que le indica a éste la recepción de una petición o *request* efectuada por el cliente. La respuesta enviada (*response*) se comunicará en la parte del cliente a través del evento *confirmation*. Este modelo se da en tecnologías como Interbus, Profibus (FMS y DP), P-NET y WorldFIP.

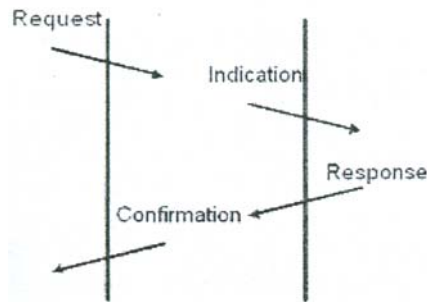


Figura 2.5. Interacción cliente/servidor típica.

Este modelo presenta problemas de consistencia en sistemas distribuidos. Si se necesita enviar un valor a dos dispositivos desde un controlador, es necesario definir dos relaciones cliente/servidor. De esta forma, los mensajes de petición tendrán que transmitirse necesariamente uno detrás de otro, por lo que el valor comunicado a los dispositivos no será el mismo ni se corresponderá con el mismo instante de tiempo.

Los sistemas distribuidos basados en buses de campo utilizan para solventar esta problemática el modelo del productor/consumidor (*producer/consumer* pero también llamado a veces *publisher/subscriber*), el cual es un modelo multipunto. Aquí las interacciones afectan a una entidad denominada productor, y a un grupo de uno o más consumidores. Los consumidores se subscriben al productor para recibir la información que necesitan. El productor del dato o variable producirá o publicará el valor en la red cada cierto tiempo según tenga programado, siendo accesible a todos los consumidores el mismo valor correspondiente al mismo instante de tiempo en el mismo mensaje de red.

Buses de campo como WorldFIP, CAN, LonWorks, EIBus, ControlNet, Swiftnet o FF utilizan este modelo, aunque hay que destacar que muchos de ellos pueden utilizar diferentes modelos simultáneamente para distintos tipos de servicio (tráfico periódico, eventos, etc.).

Los buses de campo disponen además de mecanismos de acceso al medio en el nivel de enlace que permiten “ordenar” el tráfico dentro del bus. Un aspecto importante es la planificación de mensajes, que tiene un impacto directo en la capacidad de



satisfacer los requerimientos temporales en un sistema de control, así como en la capacidad del mismo de adaptarse de forma flexible a los cambios.

2.4.3. Buses de campo estandarizados.

Hoy en día existen una gran cantidad de buses de campo en el mercado, tanto estandarizados como propietarios. En la siguiente tabla se muestra la relación entre los diferentes estándares CENELEC (Comité Europeo de Normalización Electrotécnica) e IEC (*International Electrotechnical Commission*), así como el nombre comercial por el que son más habitualmente conocidos.

| CENELEC | IEC | Nombre Comercial |
|----------------------|----------------------|---------------------|
| EN 50170-1(07/1996) | IS 61158 type 3 | P-NET |
| EN 50170-2(07/1996) | IS 61158 type 1/3/10 | Profibus |
| EN 50170-3(07/1996) | IS 61158 type 1/7 | WorldFIP |
| EN 50170-A1(04-2000) | IS 61158 type 1/9 | Foundation Fieldbus |
| EN 50170-A2(04-2000) | IS 61158 type 1/3 | Profibus-PA |
| EN 50170-A3(08-2000) | IS 61158 type 2 | ControlNet |
| EN 50254-2(10-1998) | IS 61158 type 8 | Interbus |
| EN 50254-3(10-1998) | IS 61158 type 3 | Profibus-DP |
| EN 50254-4(10-1998) | IS 61158 type 7 | WorldFIP (FIPIO) |
| EN 50325-2(01-2000) | IS 62026-3 (2000) | DeviceNet |
| EN 50325-3(04-2000) | IS 62026-5 (2000) | SDS |
| EN 50325-4(7-2002) | | CANOpen |
| EN 50295-2 | IS 6226-2 (2000) | AS-Interface |

Tabla 2.2. Relación estándares buses de campo CENELEC e IEC.

Para tener una visión global de las características de todos los buses de campo existentes en la actualidad, se muestra la siguiente tabla:



| Bus de campo | Topología | Medio Físico | Velocidad | Distancia segmento | Nodos segmento | Acceso al bus - Estándar |
|--------------|--------------------------|---|---------------------------|--------------------|----------------|--|
| ARCNET | Bus Estrella | Par trenzado Fibra óptica Coaxial | 2.5 Mbps | 122m | 255 | Paso testigo – ANSI 878 |
| AS-I | Bus Árbol Estrella | Cable de dos hilos | 167 kbps | 100m | 32 | Sondeo principal-subordinadas |
| Bitbus | Bus | Par trenzado Fibra óptica | 62.5/375 kbps 1.5 Mbps | 1200m /300m | 29 | Sondeo principal-subordinadas – IEEE 1118 |
| CAN | Bus | Par trenzado | 50 kbps a 1 Mbps | 1000m a 40m | 127 hasta 64 | CSMA/CD con arbitraje de bit – ISO 11898/11519 |
| CC-Link | Bus | Par trenzado | 156 bps a 10 Mbps | 1200m a 100m | 64 | Sondeo principal-subordinadas |
| ControlNet | Bus Árbol Estrella | Coaxial Fibra óptica | 5 Mbps 5 Mbps | 1000m 3000m | 48 | CTDMA – Basado en CAN |
| DeviceNet | Bus | Par trenzado | 125 – 250 – 500 kbps | 500 – 250 – 100 m | 64 | CSMA/CD – ISO 11898 y 11519 (basado en CAN) |
| EIB | Libre | Par trenzado Infrarrojos Red eléctrica Radio | | | | CSMA/CA – CENELEC TC105 |
| Filbus | | Par trenzado | 375 kbps | 1200m | 32 | |



| | | | | | | |
|------------------------|------------------------|--|-------------------------------|----------------|-----|---|
| Firewire | Bus Árbol | Cable de 6 hilos Fibra óptica | 100 Mbps 200 – 400 Mbps | 700m | 63 | Ranuras de tiempos – IEEE 1394 |
| Foundation Fieldbus | | Par trenzado | | 1900m | 32 | Paso testigo – ISA SP50/ IEC TC65 |
| HART | Bus | Par trenzado apantallado | 1200 bps | 3000m | 30 | Sondeo principal- subordinadas – IEC TC65 |
| Interbus-S | Anillo | Par trenzado | 500 kbps | 400m | 256 | Paso testigo – DIN E19258 |
| J1939 | Bus | Par trenzado apantallado | 250 kbps | | | CSMA/CD con arbitraje de bit – SAE J1939 (Basado en CAN) |
| LonWorks | Bus Anillo Libre | Par trenzado Fibra óptica Red eléctrica Coaxial Radio Infrarrojos | 78 kbps a 1.25 Mbps | 2700 a 130m | 64 | CSMA/CA EIA-709 |
| M3S | Bus | Cable de 6 hilos | | | | CSMA/CD con arbitraje de bit – Basado en CAN |
| M-Bus | Bus | Cable de 2 hilos | 300 a 9600 bps | 1000m | 250 | Centralizado – IEC 870; EN 1434-3 |
| Measurement Bus | Bus | Cable de 4 hilos | 110 bps a 1 Mbps | 500m | 31 | Sondeo principal- subordinadas – DIN 66348 |
| Modbus | Bus | Par trenzado | 300 bps a 19.2 kbps | 1000m | 248 | Sondeo princ-subord |



| NBIP | Bus | Cable de 2 hilos | | | | Sondeo principal-subordinadas |
|-----------|--------|--|---|---|-----|--|
| P-NET | Anillo | Par trenzado apantallado | 76.8 kbps | 1200m | 125 | Paso testigo Sondeo principal-subordinadas – CENELEC EN 50170 Vol.1; DSF 21906; DK 502058; DK 502066 |
| PROFIBUS | Bus | Par trenzado apantallado | 9.6 kbps / 19.2 kbps / 93.75 kbps / 187.5 kbps / 500 kbps | 1200m / 1200m / 1200m / 600m / 200m | 32 | Paso testigo Sondeo principal-subordinadas – CENELEC EN 50170 Vol.2; DIN STD 19245 |
| SDS | Bus | Cable de 4 hilos | 125 kbps / 250 kbps / 500 kbps / 1 Mbps | 500m | 64 | CSMA – ISO 11989 (basado en CAN) |
| SENSOPLEX | Bus | Coaxial | | | 120 | Sondeo principal-subordinadas |
| SERCOS | Anillo | Fibra óptica | 2 Mbps | | 255 | Anillo ranurado – IEC 61491 |
| SERIPLEX | Bus | Cable de 4 hilos apantallado | 98 kbps | 1500m | 300 | Sondeo principal-subordinadas |
| WorldFIP | Bus | Par trenzado apantallado Fibra óptica | 31.25 kbps 1 Mbps 2.5 Mbps 5 Mbps | 1000m 750m 500m | 32 | Centralizado (árbitro de bus) – CENELEC EN 50170 Vol. 3; NFC 46 601 a 607 |

Tabla 2.3. Comparativa de los principales buses de campo.

A continuación se describirán las redes de comunicación más importantes en la actualidad: PROFIBUS, WORLDIFIP, CAN y buses empleados en Domótica e Inmótica.

2.5. El bus de campo PROFIBUS.

PROFIBUS (PROcess Field BUS) tiene una predominante aplicación industrial, sobre todo en funciones de control y automatización. Tiene su origen en Alemania en 1987 y surgió a partir de una iniciativa de un grupo de fabricantes y usuarios con el apoyo del gobierno alemán. La norma alemana que describe el protocolo PROFIBUS es la DIN 19245, en 1996 se incluyó en la norma europea CENELEC EN 50170.

El protocolo PROFIBUS tiene tres perfiles, cada uno orientado a diferentes clases de aplicaciones: PROFIBUS-FMS, PROFIBUS-DP y PROFIBUS-PA. El perfil PROFIBUS-FMS (*Fieldbus Message Specification*) es la solución adoptada para tareas de comunicaciones industriales en el nivel superior (nivel de célula) y en el nivel de dispositivos de campo. PROFIBUS-DP (*Distributed Peripheral*) es una versión optimizada en prestaciones y dedicada especialmente a comunicaciones críticas en el tiempo entre sistemas de automatización y la periferia distribuida. Por último, PROFIBUS-PA (*Process Automation*) es la versión utilizada en automatización. Utiliza la técnica de transmisión especificada en la norma IEC 1158-2 y permite seguridad intrínseca y alimentación de las estaciones por el bus. A continuación se muestra un esquema de las variantes de PROFIBUS.

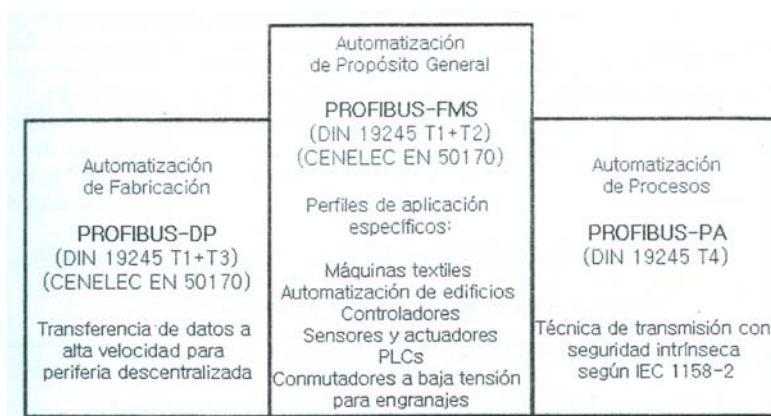


Figura 2.6. Perfiles de protocolos PROFIBUS.



PROFIBUS define las características técnicas y funcionales de un bus serie para interconectar dispositivos de campo. El sistema tiene dos tipos de estaciones denominadas principal (maestra o activa) y subordinada (esclava o pasiva). Las estaciones principales son capaces de controlar el bus y pueden transferir mensajes cuando tienen acceso a él. Las estaciones subordinadas son dispositivos periféricos más sencillos (sensores, actuadores, etc.). Éstas no tienen derechos de acceso al bus y solamente pueden confirmar o responder a mensajes enviados por la estación principal. Estas estaciones subordinadas necesitan una pequeña parte del protocolo y su realización es más sencilla.

El protocolo PROFIBUS solamente emplea tres niveles de los siete posibles en el modelo OSI: capa física, capa de enlace de datos y capa de aplicación. Además, existen otras capas para la gestión de red denominadas FMA (Fieldbus Management).

El hilo de cobre es el medio físico de transmisión básico que se utiliza para aplicaciones de fabricación. Se trata de un cable con un par de hilos trenzados y apantallados, aunque se puede usar sin apantallar si no hay excesivas interferencias electromagnéticas. Cada estación se conecta al medio físico mediante un conector de 9 patillas tipo sud-D. PROFIBUS también puede utilizar transmisión por fibra óptica para aumentar la longitud del bus y trabajar en entornos con ruido eléctrico elevado.

Se recomienda un máximo de 32 estaciones principales en la red global por razones de velocidad de respuesta, aunque para aplicaciones no críticas en el tiempo se permite un máximo de 127 (con direcciones desde 0 a 126).

PROFIBUS es un bus de campo que puede trabajar a velocidades bastante elevadas, alcanzando los 12 Mbps, por lo que es muy apropiado en aplicaciones donde las comunicaciones en tiempo real son importantes.



2.6. El bus de campo WORLDVIP.

WorldVIP (*World Factory Instrumentation Protocol*) es uno de los sistemas más utilizados en la industria actual, sobre todo en Francia que es donde surgió. Existe un gran número de empresas que fabrican dispositivos y equipamiento dotados con interfaces de comunicación WorldVIP. Este protocolo ha sido impulsado por los fabricantes franceses (Cegelec, Telemecanique, Efisysteme, Gespac, etc.). En 1996 fue recogido por el CENELEC en su norma EN 50170.

Es un bus de campo diseñado para establecer comunicaciones entre el nivel de sensores/actuadores y el nivel de unidades de proceso (PLCs, controladores, etc.) en los sistemas de automatización. Para definir esta red industrial se han tenido en cuenta consideraciones económicas, como la de reducir el coste del cableado y ahorrar en tiempo de diseño e instalación. También se consideraron aspectos técnicos como: un fácil mantenimiento y ampliación, simplificación del tradicional cableado punto a punto, garantizar un tiempo de respuesta, conseguir una seguridad aceptable y que las variables tuvieran un grado de accesibilidad elevado.

WorldVIP puede ser usado en todo tipo de arquitecturas (centralizada, descentralizada, principal-subordinado) pudo un algoritmo ejecutarse en una única unidad de proceso o estar distribuido entre varias unidades de proceso en diferentes nodos de la red. El mecanismo de difusión (*broadcast*) empleado en este tipo de red garantiza la consistencia espacial y temporal de los datos para todas las estaciones que consumen un conjunto de variables.

El protocolo WorldVIP tiene tres niveles, de los siete posibles en el modelo OSI. Los dos primeros son el nivel físico y el nivel de enlace, y el último es el nivel de aplicación. Sin embargo también tiene definida una función de gestión de red que se comunica con los anteriores tres niveles.

El medio de transmisión puede ser un cable de pares trenzados apantallados o fibra óptica y presenta topología de bus lineal. Además del cable que conforma el bus, en una red WorldVIP se pueden distinguir los siguientes dispositivos:

- JB (*Junction Box*): permite hasta dos accesos para derivaciones.
- TAP: punto de conexión al cable del bus donde puede conectarse un dispositivo de campo.
- REP (*Repeater*): amplifica la señal entre dos tramos del bus para conseguir longitudes mayores.
- DB (*Diffusion Box*): sirve para unir varios segmentos terminales al cable principal del bus.
- DS (*Disconnectable Subscriber*): abonados o nodos que son desconectables localmente.
- NDS (*Non-locally Disconnectable Subscriber*): abonados o nodos que no se pueden desconectar localmente.

El bus se puede dividir en segmentos que se interconexionan mediante repetidores, pudiendo haber un máximo de 32 dispositivos conectados en cada segmento, y un máximo de 256 dispositivos en toda la red. La topología física limita a 3 el máximo número posible de repetidores entre 2 estaciones cualesquiera. Una caja de distribución (DB) puede servir para conectar hasta 8 estaciones a un único punto de conexión.

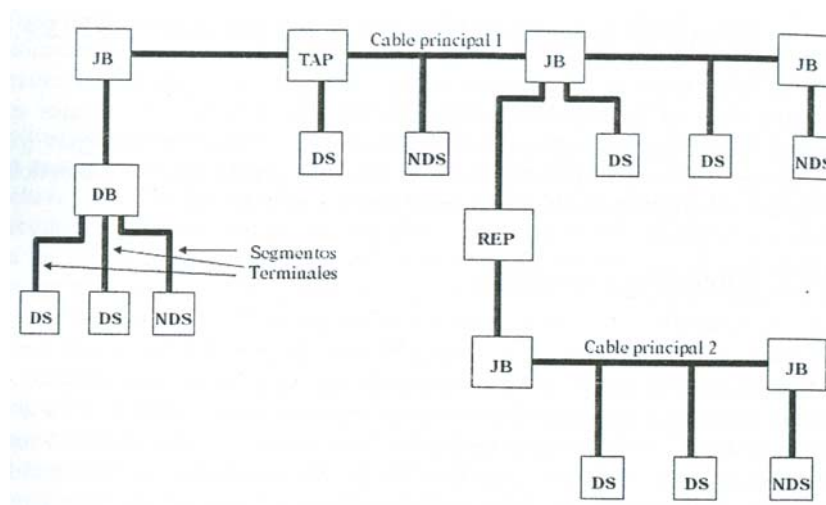


Figura 2.7. Ejemplo de estructura física de una red WorldFIP.



En el protocolo WorldFIP se han definido tres velocidades posibles para la transmisión de mensajes sobre un medio físico de cable de cobre:

- 31,25 kbps (baja velocidad). Para aplicaciones especiales.
- 1 Mbps (alta velocidad). Es la velocidad estándar.
- 2,5 Mbps (alta velocidad). Para aplicaciones especiales.

Para fibra óptica se definió una velocidad de 5 Mbps.

Según sea la velocidad de funcionamiento del bus, las distancias máximas posibles entre dos estaciones cualesquiera de la red varían. Si no se usan repetidores y considerando un medio físico de cable de cobre de pares trenzados apantallados, las distancias máximas de una red WorldFIP son:

- 1.900 metros a 31,25 kbps.
- 750 metros a 1 Mbps.
- 500 metros a 2,5 Mbps.

Usando repetidores se pueden multiplicar las anteriores distancias por cuatro ya que el número máximo de repetidores permitido entre dos estaciones es de tres.

Los bits que forman las tramas transmitidas por la capa de enlace de datos son codificados por el nivel físico usando el código Manchester. Con este código es posible transmitir simultáneamente los datos y el reloj para la sincronización de señales necesaria en WorldFIP, ya que es un bus de campo serie síncrono. Todas las tramas de este protocolo (tramas de pregunta, respuesta, mensajes, etc.) se componen de tres partes: una secuencia de inicio de trama, un campo de datos y control, y una secuencia de fin de trama.

El protocolo WorldFIP basa su funcionamiento en dos elementos claves: variables y mensajes. Da preferencia a la transmisión de variables sobre la de mensajes, asegurando la transmisión de éstas con una periodicidad determinada. Esto es muy útil



en aplicaciones en tiempo real en procesos industriales donde la consulta de cierta información es crítica para la seguridad del sistema.

En conclusión, se puede decir que WorldFIP tiene un alto grado de eficiencia y puede transmitir en tiempo real una gran cantidad de información, siendo muy útil entre el nivel de sensores/actuadores y el nivel de las unidades de proceso en aplicaciones de control industrial y automatización.

Además, WorldFIP está considerado como uno de los pocos buses de campo capaces de ofrecer una conexión transparente a Internet. Por este motivo, la asociación WorldFIP anunció en el año 2000 su intención de desarrollar una evolución del protocolo WorldFIP para convertirse en un protocolo de bus de campo para Internet.

2.7. El bus de comunicaciones CAN.

El bus de comunicaciones CAN (*Controller Area Network*) surgió en la empresa Robert Bosch GmbH, a finales de los 80, de la necesidad de comunicar las distintas unidades electrónicas existentes en el automóvil, tales como frenado, ABS, inyección de combustible, etc. Posteriormente su utilización se ha ido expandiendo a otros campos de aplicación distinta de la automoción y han aparecido buses de campo basados en CAN tales como DeviceNet (con fuerte implantación en la automatización industrial) y CANopen.

El bus CAN es utilizado en el automóvil como bus de comunicaciones de lo que se denomina sistemas distribuidos embebidos, es decir para unir unidades de control electrónicas que están “embebidas” o “empotradas” en los sistemas que controlan. En un automóvil pueden existir varias redes CAN, aunque lo normal es que sean dos, una de alta velocidad de transmisión y otra de baja velocidad. La red CAN de alta velocidad enlaza unidades de control en tiempo real como las de inyección de combustible, frenado ABS o unidad de encendido en motores de encendido provocado. La red CAN de baja velocidad enlaza dispositivos electrónicos tales como los elevalunas, conjuntos de luces, sistema de climatización, etc. Cifras de 97 millones de controladores CAN vendidos en 1998 hablan por sí solas del volumen de nodos CAN que se fabrican cada



año, muchísimo superiores a las de cualquier otro bus de comunicaciones en la actualidad. Este éxito es debido a que dichos elementos presentan importantes ventajas como son un bajo coste, fiabilidad y capacidad de funcionar en entornos agresivos. Estas características, unidas a las de robustez, facilidad de uso y alto grado de capacidad de tiempo real, hicieron que el bus CAN pronto se utilizara en aplicaciones industriales, tanto en sistemas distribuidos embebidos (robots, máquinas textiles, máquinas empaquetadoras), como en sistemas abiertos para automatización, siendo soporte de varios buses de campo como CANOpen, DeviceNet, SDS, CAN-Kingdom, etc.

Otros campos de aplicación del bus CAN son la automatización naval (por ejemplo para el control distribuido de salas de máquinas desatendidas), como subred en edificios inteligentes, controlando sistemas tales como los ascensores, sistemas de aire acondicionado, etc., en el mundo del ferrocarril (control de puertas en vagones) y en el mundo de la aviación para unir sensores de estado de vuelo y sistemas de navegación. En el campo de la medicina CAN se ha utilizado como red embebida en equipos médicos como equipos de rayos X, *scanners*, etc.

También hay que considerar la existencia de muchos interfaces hardware y paquetes software para facilitar el análisis del bus CAN de empresas como VECTOR y National Instruments, la existencia de placas de bajo costo que proveen soporte para construir prototipos rápidamente e incluso los fabricantes de osciloscopios comienzan a incluir modos de disparo apropiados al bus CAN.

El bus CAN está muy bien adaptado para interconectar dispositivos de entrada-salida, sensores y actuadores inteligentes. Además esta comunicación la realiza con un buen comportamiento en tiempo real y de un modo muy seguro dada la sofisticada detección y confinamiento de errores que utiliza.

Se puede afirmar por tanto que entre los buses de tiempo real aplicables a sistemas distribuidos en los que la relación prestaciones/coste es determinante, puede decirse que hasta la fecha ninguno ha alcanzado la combinación de aceptación, disponibilidad



de dispositivos de diversos fabricantes y robustez que ofrece el bus CAN. Todas estas consideraciones han hecho que el abanico de campos de aplicación donde se ha introducido el bus CAN sea cada vez más amplio, incrementándose el uso de buses de campo basados en CAN como DeviceNet y CANopen. Por último en el año 2001 Bosch ha especificado el protocolo TTCAN (*Time Trigered CAN*) donde se intenta acercar esta nueva versión a las características necesarias que ha de tener un bus de comunicaciones en sistemas de tiempo real estricto (todas las acciones deben terminar dentro del plazo especificado).

2.8. Buses y protocolos en Domótica e Inmótica.

El grado de desarrollo actual de la Domótica e Inmótica en España y el mundo, es considerable sobre todo si se tiene en cuenta su reciente historia (para encontrar sus orígenes habría que remontarse a finales de los años 80 y principios de los 90). A nivel físico, las redes domóticas pueden clasificarse según la siguiente tabla:

| Tipo | Usabilidad | Características y requerimientos |
|--|--|---|
| Transmisión con cable | | |
| Cableado dedicado | Muy fácil, muy extendido, económico | Permiten crear grandes redes de equipos |
| Par trenzado | Proviene de usos industriales | Gran seguridad de transmisión |
| Cable coaxial | Utilizado en el envío de señales de video. Bastante implantado | Inmune a interferencias pero muy rígido para instalación |
| Red eléctrica instalada | No necesita instalación adicional de cableado | Poca seguridad y velocidad. Ventaja de aprovechar la instalación eléctrica instalada. |
| Fibra óptica | Gran capacidad | Se utiliza para transmitir gran cantidad de información |
| Transmisión sin cable por radiofrecuencia | | |
| Bluetooth V1 y 2 | Bastante extendido | Es un estándar. Velocidad de transmisión media y corto alcance |
| IEEE 802.11b | Bastante extendido | Es un estándar, admiten velocidades altas de transmisión |
| IEEE 802.11g | Poco extendido | Altísimas velocidades de transmisión en frecuencia estándar |
| IEEE 802.15.4 | Poco extendido | Es un estándar, velocidades de transmisión bajas, pensado para dispositivos de gestión de edificios |
| IEEE 802.16 a, b, c | Poca implantación | Es un estándar, para redes inalámbricas metropolitanas y redes entre edificios |

Tabla 2.4. Medios físicos de las redes domóticas.



A nivel mundial, existen dos estándares para domótica e inmótica: KNX-EIB (en Europa) y SCP (en EEUU).

La asociación KNX-EIB nace como iniciativa de tres organizaciones, que son:

- EIBA (*European Installation Bus Association*), representante del sistema EIB.
- BCI (*BatiBUS Club International*), representante del sistema Batibus.
- EHSA (*European Home System Association*), representante de la tecnología EHS.

Se trata de que, partiendo de los sistemas EIB (*European Installation Bus*) y BatiBUS, crear un único estándar europeo que sea capaz de competir en calidad, prestaciones y precios con otros sistemas norteamericanos como LonWorks o CEBus, y finalmente con el estándar americano de convergencia SCP (*Simple Control Protocol*). Actualmente la asociación Konnex está terminando las especificaciones del nuevo estándar (versión 1.0) el cual será compatible con los productos EIB instalados. Este nuevo estándar tendrá lo mejor del EIB, del EHS y del Batibus.

Respecto al nivel físico, el sistema puede funcionar sobre los siguientes medios:

- Par trenzado a 9600 bps. Además por estos dos hilos se suministra 24 Vdc para la telealimentación de los dispositivos.
- Corrientes portadoras sobre 230 Vac/50 Hz (powerline) a 1200/2400 bps. La distancia máxima que se puede lograr sin repetidor es de 600 metros.
- Usando el estándar Ethernet a 10 Mbps (IEC 802-2). Permite la transferencia de telegramas a través del protocolo IP a instalaciones remotas.
- Radiofrecuencia: usando varias portadoras, se consiguen distancias de hasta 300 metros en campo abierto. Para mayores distancias o edificios con múltiples estancias se pueden usar repetidores.
- Infrarrojo: para el uso con mandos a distancia en salas o salones donde se pretenda controlar los dispositivos EIB instalados.



En la práctica, sólo el par trenzado ha conseguido una implantación masiva mientras que los demás apenas han conseguido una presencia testimonial.

El SCP es un intento de Microsoft y de General Electric de crear un protocolo para redes de control que consiga afianzarse como la solución en todas las aplicaciones de automatización de edificios. Se trata de poner un poco de orden en la oferta existente en EEUU en este ámbito (X-10, CEBus, LonWorks y otros) y auspiciar la convergencia de todos estos hacia un protocolo abierto, además de desarrollar un conjunto de productos que cubran todos los requisitos de automatización de las instalaciones.

A nivel físico el SCP ha escogido una solución basada en la transmisión de datos por las líneas de baja tensión (ondas portadoras) que ya estaba desarrollada, el CEBus. En EEUU, donde llevan varios años de adelanto en la implantación de sistemas domóticos e inmóticos respecto a Europa, el X-10, en el mercado residencial y el Lonworks, en el mercado profesional, tienen copado el mercado.

Actualmente las empresas Domsys, ITRAN Communications Ltd y Mitsubishi Electric Corporation, están desarrollando circuitos integrados que implementen la especificación SCP en poco espacio y a bajo coste, haciendo posible su uso en multitud de dispositivos eléctricos. Está previsto el desarrollo de varios medios físicos adicionales como el par trenzado y la radiofrecuencia.

El SCP está optimizado para su uso en dispositivos eléctricos y electrónicos que tienen una memoria y una capacidad de proceso muy limitadas. Al igual que otros buses o protocolos de control distribuido, el SCP está diseñado para funcionar sobre redes de control con un ancho de banda muy pequeño (<10 kbps) y optimizado para las condiciones de ruido características de las líneas de baja tensión,

En el futuro, los dispositivos SCP usarán modelos definidos por el UPnP (*Universal Plug&Play*) que serán configurados mediante el acceso a un conjunto de primitivas o APIs (*Application Program Interface*). Se trata de asegurar la conexión punto a punto entre dispositivos y definir un conjunto de funciones distribuidas extremo a extremo que permita el desarrollo de múltiples servicios con un bajo coste y de manera segura.



Ahora bien, a parte de los dos estándares anteriormente comentados (KNX-EIB y SCP), existen sistemas propietarios (es decir, sistemas pensados para trabajar en el entorno de los productos del mismo fabricante, no admitiendo comunicación con otros de la competencia).

Los niveles físicos de transmisión de datos entre sensores y actuadores suele depender de la arquitectura de conexionado. Así en los de arquitectura centralizada se utiliza normalmente cableado punto a punto (con cable convencional), en su defecto utiliza la instalación eléctrica convencional con el empleo de la tecnología de portadoras sobre red eléctrica o finalmente de manera más excepcional radiofrecuencia. En los de arquitectura distribuida se utiliza un bus físico por cable de características dadas por el fabricante.

Centrándonos en las instalaciones domóticas industriales, se distinguen tres niveles bien diferenciados, como son el nivel de campo, de automatización y de gestión. La arquitectura en cada uno de ellos es distribuida. Si el tamaño de la instalación es pequeño se resolverá la instalación con el nivel de campo. Si se aumentan las necesidades serán necesarios dos niveles (campo y automatización) y si la instalación es lo suficientemente compleja se necesitará inclusive el nivel de gestión.

Como es lógico, el mercado de la Inmótica tiene su punto de partida en el entorno industrial, de hecho los productos empleados son más robustos, fiables y costosos que los empleados en el mercado domótico. Esta es la razón por la que es un sector ciertamente monopolizado a nivel mundial y existen detrás de estas tecnologías, grandes multinacionales del sector de la automatización y el control. Algunos de las más significativas son: Siemens, Honeywell, el Grupo Scheneider o Johnson Controls.

2.9. Otros tipos de comunicación.

En este punto se comentarán brevemente dos interfaces de comunicación: MPI (Multi Point Interface), que puede ser utilizado en este proyecto como alternativa a Ethernet

Industrial en la comunicación PC - Autómata, y el bus USB, ampliamente empleado en los entornos informáticos.

Si optamos por la interfaz MPI para la comunicación PC-Autómata, el esquema de conexiones quedaría de la siguiente manera:

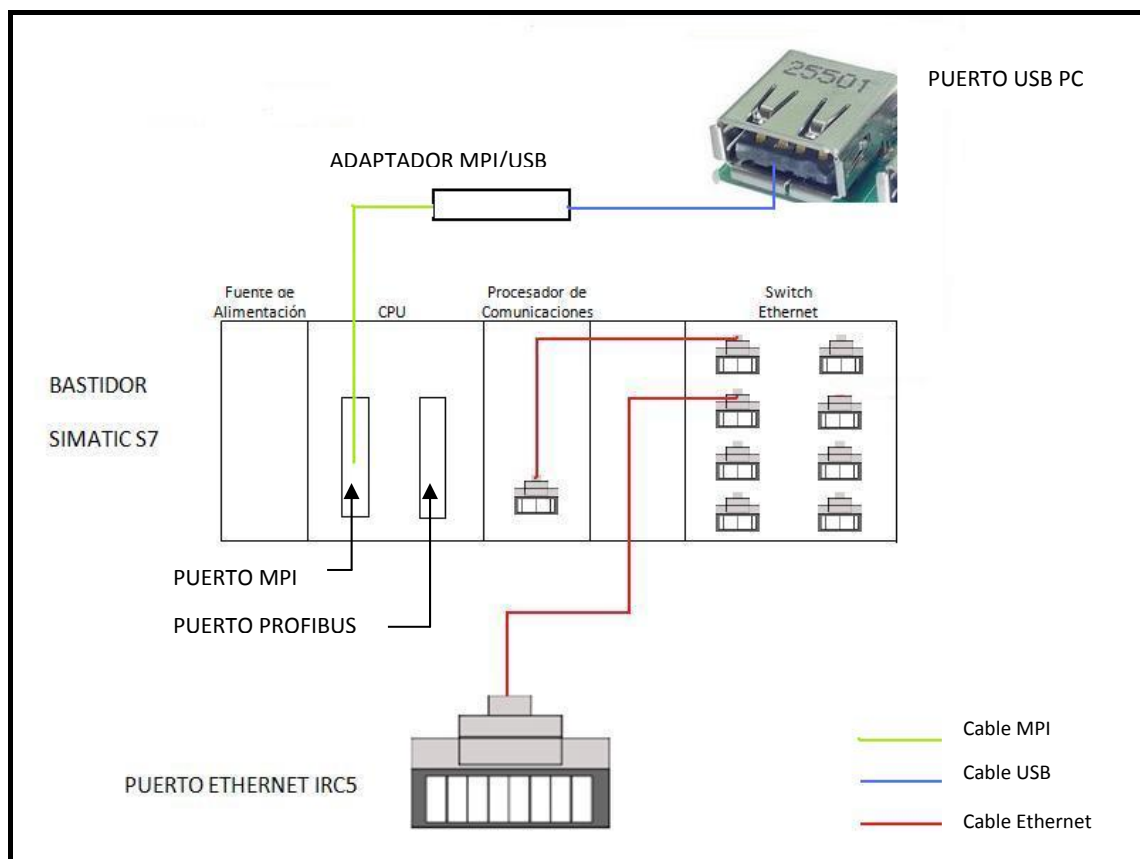


Figura 2.8. Esquema de conexiones; interfaz Industrial Ethernet y MPI.

MPI es un protocolo de comunicaciones específico de SIEMENS, con lo que todas las CPUs de PLCs SIEMENS incorporan una interfaz de comunicación de este tipo. Este hecho propicia que la comunicación a través de dicho interfaz sea muy sencilla.



El interfaz MPI ha sido concebido como puerto de programación (a través del cual se cargan los programas en la CPU del PLC) y para interconectar pocas CPUs con el fin de intercambiar cantidades de datos reducidas, siendo el número máximo de estaciones activas en la red de 32, y la velocidad máxima de transferencia de 187,5 KBits/seg.

En MPI son posibles varios interlocutores. La conexión con la red de comunicación se efectúa en el PLC a través de la interfaz MPI de la CPU, y en WinCC a través de la interfaz MPI instalada.

Por MPI la comunicación se realiza directamente con la CPU del autómatas mientras que por Ethernet es necesario un módulo procesador de comunicaciones (CP).

Es preferible emplear la tecnología Ethernet debido a su mayor rapidez en el manejo de datos y a sus reducidos costes de instalación, configuración y mantenimiento.

Por su parte el bus USB, definido en 1995 y revisado en el 2000, presenta las siguientes características:

- Velocidad de transmisión de 1,5 Mb/s, 12 Mb/s o 480 Mb/s en la versión 2.0.
- Conexión física de hasta 127 dispositivos.
- Cable de bus apantallado con 4 hilos, par trenzado (para minimizar interferencias) para datos y 2 hilos de alimentación: 5 voltios y masa con 500 mA máximo para alimentar a periféricos de bajo consumo, tales como un teclado o un ratón.
- Transmisión diferencial sobre las líneas de datos.
- Máxima distancia entre el computador maestro o host y el dispositivo USB de 5m sin repetidores (hubs), que se puede aumentar hasta 25m con los hubs USB disponibles comercialmente.
- Conexión y desconexión de dispositivos “en caliente”, no es necesario apagar el ordenador “host”.



- Soporte “plug and play”, de manera que el computador maestro puede identificar al dispositivo USB al conectarlo físicamente en el bus y configurarlo cargando el driver apropiado.

USB se ha convertido en el bus estándar para la conexión de periféricos a un PC. Además los puertos USB no solo están disponibles en los PC de sobremesa y portátiles, sino también en PCs industriales, PCs monoplaca y pantallas táctiles de interfaz hombre-máquina (*Human Machine Interface* HMI). Entre sus mayores ventajas están su utilización masiva y la facilidad de empleo, pues una vez cargado el software del driver del dispositivo USB en el host, sólo hay que conectar físicamente el periférico al bus en cualquiera de los puertos USB disponibles para que el host reconozca el periférico. Las ventajas anteriores han hecho atractiva su utilización en la industria. En el campo de los sistemas de adquisición de datos están apareciendo sistemas basados en bus USB, por ejemplo National Instruments dispone de los dispositivos USB de la serie M que toman su alimentación del propio bus, disponen de separación galvánica y existen distintos tipos para medida de temperatura, presión, flujo y vibración. Este tipo de dispositivos USB tienen su aplicación sobre todo en sistemas de adquisición de datos portátiles poco distribuidos ya que como se comentó la máxima longitud de un cable USB es 5 metros sin repetidores.

2.10. Situación actual y tendencias.

Como suele ser habitual en todas las áreas de la técnica, cuando una tecnología ha alcanzado la madurez, surgen nuevas aplicaciones con mayores requerimientos que implican la necesidad de nuevos desarrollos para satisfacerlos. En el área de los buses de campo, la necesidad de proporcionar mayores velocidades de transmisión, mejor soporte a los requerimientos de tiempo real, o mejorar la flexibilidad y despliegue introducen la necesidad de nuevos protocolos, medios de transmisión, y software de gestión.



Estas nuevas tendencias en el área de los buses de campo vienen marcadas por las siguientes necesidades:

- Cambios en los escenarios de aplicación.
- Factores tecnológicos y de mercado.
- Integración vertical.
- Incremento de la complejidad.
- Seguridad.

Desde sus orígenes, los buses de campo tenían como objetivo la reducción del cableado de los sistemas centralizados tradicionales en esa época, así como la dotación de más inteligencia en los dispositivos de campo, pasando así a ser sistemas distribuidos. Sin embargo, el uso de este tipo de redes se ha ampliado a otros nuevos escenarios, con ciertas similitudes con el planteamiento inicial pero con nuevos retos que resolver. El caso más representativo de este hecho es las redes utilizadas en la domótica. Si bien la domotización de un hogar no difiere mucho en los aspectos técnicos de las redes de comunicaciones que han de dar soporte a la comunicación entre sensores y actuadores inteligentes y el centro de control y comunicaciones de éste hacia afuera, la domotización de edificios enteros presenta una problemática bien diferente. Esto es debido al importante incremento de la complejidad de los sistemas, que pasan a tener cientos, e incluso miles de sensores y actuadores distribuidos por el edificio. Este incremento de la complejidad introduce la necesidad de nuevas herramientas de configuración y programación de redes industriales que faciliten esta complicada tarea.

Otro hecho fundamental en las nuevas tendencias en este campo es la fuerte penetración de Ethernet como nivel físico de varias propuestas de buses de campo. El uso de Ethernet e Internet permiten un salto adelante en la integración vertical de la empresa. A través de Internet, desde cualquier parte del mundo es posible visualizar, configurar y modificar el comportamiento de los dispositivos inteligentes de una red



industrial. Hoy en día existen circuitos integrados que incluyen en una misma pastilla microcontroladores, memoria, puerto Ethernet y la pila IP², permitiendo el desarrollo de sensores y actuadores de bajo coste que pueden ser configurados y consultados mediante http.

Finalmente, es necesario destacar el surgimiento del requerimiento de seguridad en este ámbito. Este concepto, muy extendido en el mundo de las redes de ordenadores e Internet, no fue tenido en cuenta en los momentos iniciales del desarrollo de los buses de campo. El hecho de ser, en sus inicios y en todos los casos, redes cableadas y cerradas hacía que la seguridad no fuera un hecho preocupante. Sin embargo, tanto la cada vez mayor penetración de las redes inalámbricas, como la introducción de la pila IP en los dispositivos de campo, han generado que esta problemática sea un tema de especial interés en la actualidad. Hay que recordar la importancia de la información que estos equipos manejan (tanto para la seguridad de las personas, como de las empresas, así como económica puesto que el normal funcionamiento de la producción industrial depende de estos equipamientos), por lo que la apertura al mundo que supone Internet y las redes inalámbricas ha hecho que la seguridad haya pasado a tener una importancia vital.

² La pila IP constituye un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. El protocolo IP realiza las tareas básicas para conseguir transportar datos desde un origen a un destino.



Capítulo 3:

Descripción del Sistema.

3.0. Condiciones de partida.

Como se ha explicado en la introducción, en la aplicación desarrollada intervienen tres elementos hardware: un PC, un autómatas Siemens S7 y el controlador de robot ABB IRC5. Todos ellos están interconectados mediante la interfaz Ethernet Industrial, que posibilita el envío y recepción de datos en los dos extremos de la red, tanto en el PC como en el controlador IRC5. Podríamos decir que el autómatas es el hardware que actúa de intermediario entre el PC y el controlador IRC5.



Figura 3.1. Diagrama de bloques del sistema.

La implementación de las comunicaciones desarrollada en el proyecto podría dividirse en dos fases:

1. Comunicación PC – Autómatas (véase punto 4.1).
2. Comunicación Autómatas – Controlador de robot (véase punto 4.2).

En la primera fase, disponemos de un PC con tarjeta de red Ethernet y de un autómatas. Este elemento hardware incluye, además de su propio *firmware*, un módulo procesador de comunicaciones y un módulo Switch, trabajando ambos mediante interfaz Ethernet Industrial. Gracias a estos dos módulos, y desde el software WinCC instalado en el PC, se podrá acceder a la memoria de datos del autómatas. Podremos activar bits/bytes en el mismo y leer datos almacenados en él. De esta forma se consigue una comunicación bidireccional entre el PC y el autómatas.



También en esta primera fase, se creará la red Ethernet entre el PC y el autómatas. Para ello se emplea el software de Siemens Simatic Manager, mediante el cual se asigna una dirección IP coherente para cada elemento presente en la red.

En la segunda fase, comunicación Autómatas – Controlador de robot, básicamente se realizan cuatro operaciones:

1. Se integra el controlador del robot a la red Ethernet PC - Autómatas previamente creada. Esta integración se efectúa a través del puerto Ethernet interno del IRC5, que se conecta con el procesador de comunicaciones del autómatas a través del Switch Ethernet Industrial de éste.
2. Creación de enlace TCP en el autómatas. A través de este protocolo, y mediante el direccionamiento IP, será posible la comunicación con el programa RAPID implementado en el controlador de robot.
3. Programación del autómatas mediante el software Step7 Simatic Manager. Consiste básicamente en parametrizar adecuadamente los bloques de programa que realizan las funciones de envío y recepción de datos en el autómatas.
4. Programación del controlador de robot en RAPID por interfaz *sockets*. Se trata de un canal de comunicación definido por direcciones IP, un protocolo (en nuestro caso TCP) y un número de puerto. A través de esta interfaz, primero se localiza al interlocutor (el autómatas) y luego se envían y reciben datos de éste.

A continuación se describen los tres elementos hardware citados, así como el software y los lenguajes utilizados para su programación.

3.1. Descripción del hardware.

3.1.1. PC



Figura 3.2. PC – SCADA.

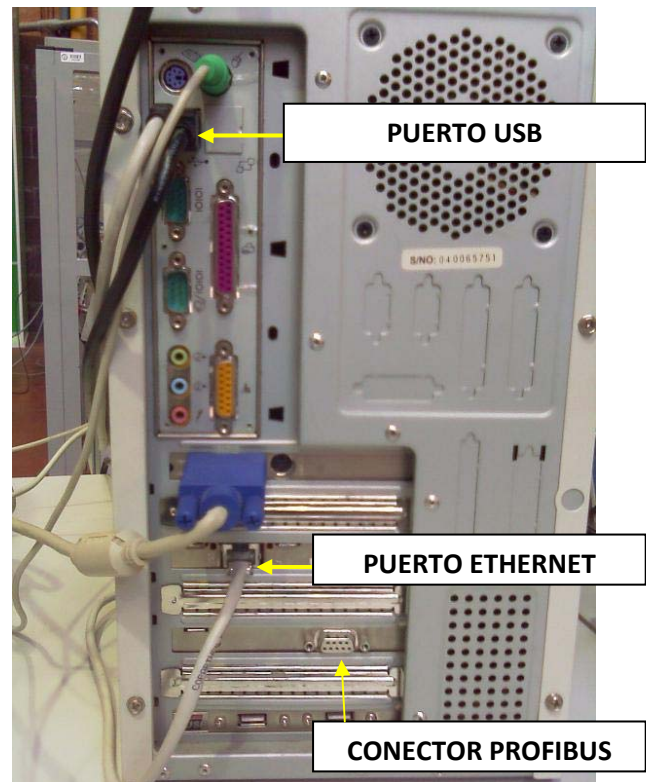


Figura 3.3. Puertos de comunicación del PC.

- Tarjeta de red NIC Fast Ethernet PCI Familia RTL8139 de Realtek.
- Tarjeta PROFIBUS.
- Procesador AMD Athlon XP 2600+
- RAM 523.760 KB
- Sistema Operativo Windows 2000 Professional SP4
- Resolución de pantalla 1024x768
- Adaptador USB a MPI (*Multi Protocol Interface*). Si, en lugar de Ethernet Industrial, se desea emplear la interfaz MPI de la CPU del autómatas para comunicarse con el PC (por USB), es necesario este adaptador.



Figura 3.4. Adaptador USB a MPI.

Como puede observarse, la comunicación del PC con el autómata podría haberse realizado, en lugar de a través de Ethernet Industrial, por PROFIBUS. La principal razón por la que se eligió la interfaz Ethernet Industrial es su bajo coste. Ethernet únicamente implica costes del cableado, insignificantes frente al coste de la tarjeta PROFIBUS que habría que instalar en el controlador de robot ABB IRC5.

Además, otras ventajas que presenta Ethernet Industrial frente a PROFIBUS son una mayor velocidad de transmisión de datos, mayor robustez y la posibilidad de comunicación por protocolo TCP/IP con equipos ajenos (como por ejemplo el controlador de robot IRC5).

3.1.2. Autómata Siemens S7 314C.

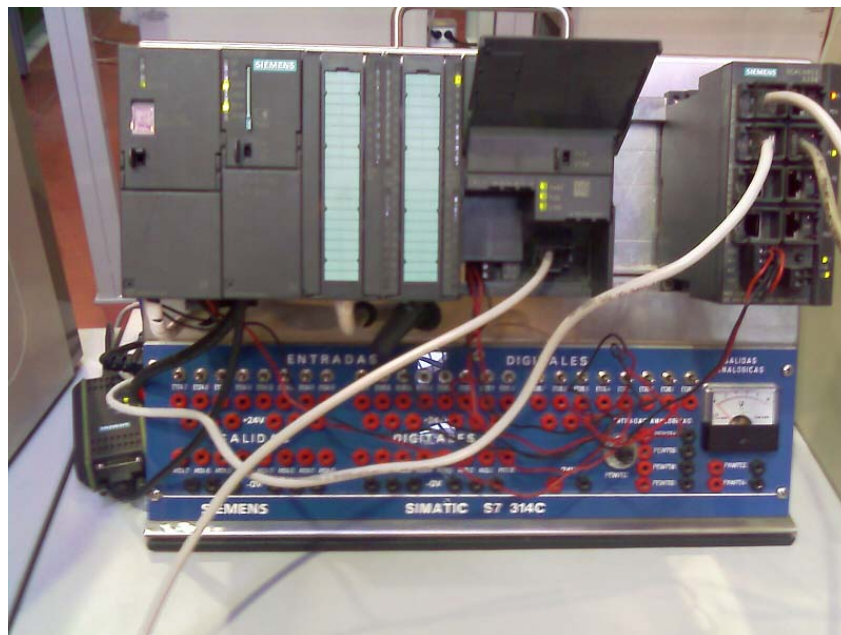


Figura 3.5. Autómata Siemens S7 314C.

El autómatas se compone de los siguientes módulos:

- Fuente alimentación PS 307 2A: 120/230V AC;24VDC/2A
- CPU 314C-2 DP: memoria de trabajo 96KB; 24 señales digitales de entrada; 16 señales digitales de salida; 5 entradas analógicas; 2 salidas analógicas integradas; 4 salidas de impulsos (2,5kHz);contaje y medida 4 canales con encoders incrementales 24V (60kHz);función de posicionamiento integrada; conexiones MPI + DP (maestro DP o esclavo DP); configuración en varias filas de hasta 31 módulos; emisor y receptor para comunicación directa; equidistancia; routing; comunicación S7 (FBs/FCs cargables); firmware V2.0.
- Procesador de Comunicaciones CP 343-1 Advanced-IT: CP S7 para Industrial Ethernet, PROFINET IO Controller, ISO y TCP/IP con interfaz SEND-RECEIVE y FETCH-WRITE; datos largos; UDP; TCP; ISO; Comunicación S7; Routing; cambio de módulo sin PG; con servidor de Web y E-Mail; 10/100 Mbit; inicialización vía LAN; IP-Multicast; sincronización horaria por procedimiento SIMATIC o NTP; protección de acceso por lista de acceso IP, FTP, SNMP, DHCP, HTML Diagnóstico; archivo de datos en C-PLUG, PROFINET CBA, Firmware V1.1.
- Switch Industrial Ethernet SCALANCE X208: 8 puertos 10/100 Mbit/s RJ45; diagnóstico por LED; contacto de señalización de fallo con pulsador set; alimentación redundante, profinet io-device; gestión de red; gestor de redundancia integrado.



Figura 3.6. Switch Industrial Ethernet SCALANCE X208.

3.1.3. Controlador IRC5 de robot ABB IRB 2400.

| Interfaces | |
|---|---|
| Entradas/salidas | Hasta 2.048 señales |
| Digitales | 24 V CC o señales de relé |
| Analógicas | 2 x 0-10V, 3 x $\pm 10V$, 1 x 4-20mA |
| Canales en serie | 3 x RS 232/RS 422 |
| Red | Ethernet (10/100 Mbits por segundo) |
| Dos canales | Servicio y LAN |
| Escanners Fieldbus | DeviceNet Interbus Profibus DP Entrada/Salida remota Allen-Bradley |
| Conexiones para señales con el brazo superior del manipulador | |

Tabla 3.1. Parámetros característicos del controlador IRC5.

- Hardware de control: sistema multiprocesador; Bus PCI; UCP Pentium; Disco flash o disco duro para memoria masiva; Alimentación de reserva para utilización en caso de corte de corriente; Interfaz para memoria USB.
- Software de control: diseño orientado al objeto; lenguaje de programación de robots RAPID, de alto nivel; Portátil, abierto, ampliable; formato de archivo PC-DOS.
- Tamaño 970 x 725 x 710mm.
- Peso 150 Kg.

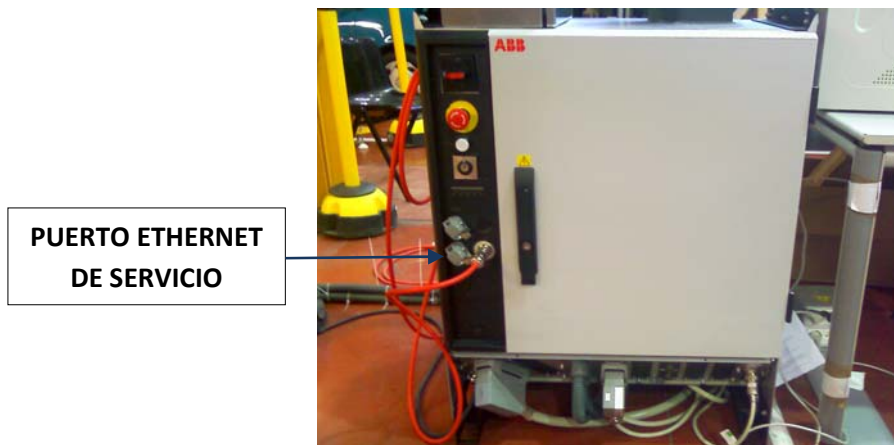


Figura 3.7. Controlador IRC5; Puerto Ethernet externo (servicio).

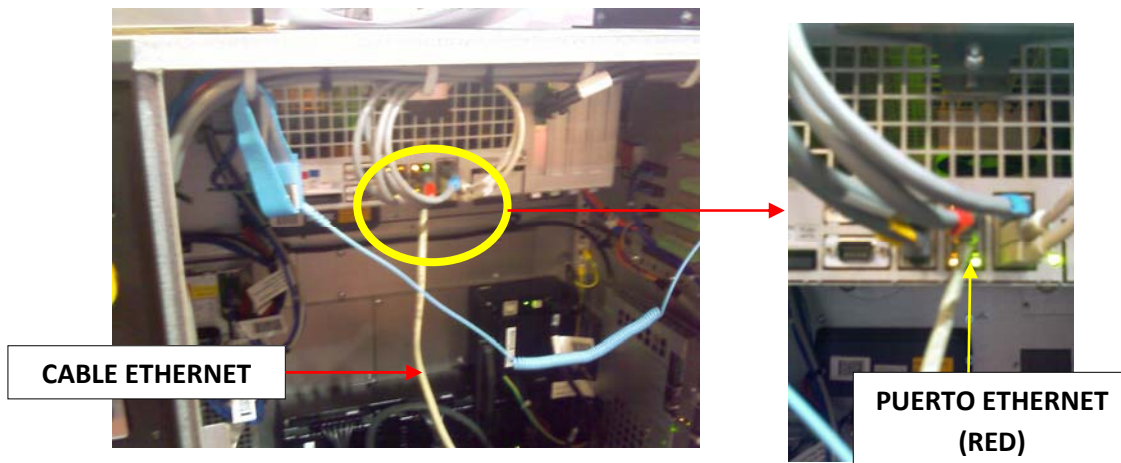


Figura 3.8. Controlador IRC5; Puerto Ethernet interno (red LAN).

Aunque el robot en sí mismo, a nivel físico, no es un aspecto relevante en lo que a las comunicaciones se refiere, a continuación se muestran sus parámetros fundamentales.

- Grados de libertad del manipulador: 6.
- Repetibilidad de posición: 0.06 mm.
- Alimentación 200–600 V, 50/60 Hz.
- Señales de alimentación integradas: 23 a 50 VDC, 10 a 250 VAC.
- Máxima presión del aire: 8 bares.

- Temperatura de trabajo 5°C - 45°C.
- Peso 380 Kg.



Figura 3.9. Robot ABB IRB 2400.

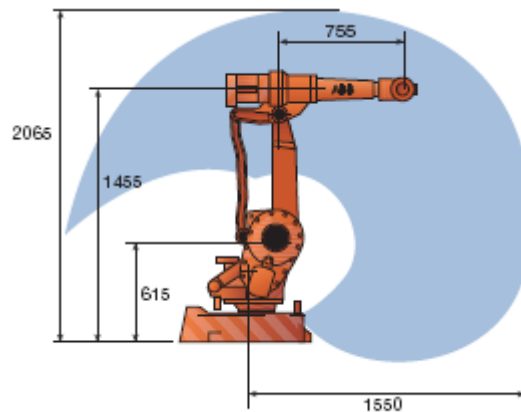


Figura 3.10. Dimensiones del robot.

3.1.4. Cable de red Ethernet conector RJ45.



Figura 3.11. Cable Ethernet, conectores RJ45.

Como puede verse en el esquema de conexiones del punto 4.3.1, serán necesarios tres cables Ethernet para:

- En el autómata, conexión del procesador de comunicaciones CP 343-1 con el Switch Ethernet SCALANCE X208.
- Conexión de Switch Ethernet SCALANCE X208 con tarjeta de red del PC.
- Conexión de Switch Ethernet SCALANCE X208 con tarjeta de red del IRC5.

Los tres cables utilizados son directos, pensados para conectar dispositivos desiguales como son una tarjeta de red con un switch. En este tipo de cables ambos extremos del cable deben tener la misma distribución.

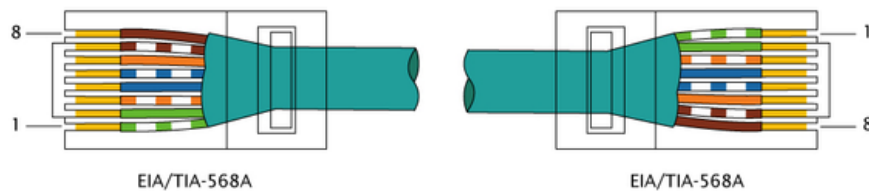


Figura 3.12. Cable directo; Distribución 568A.

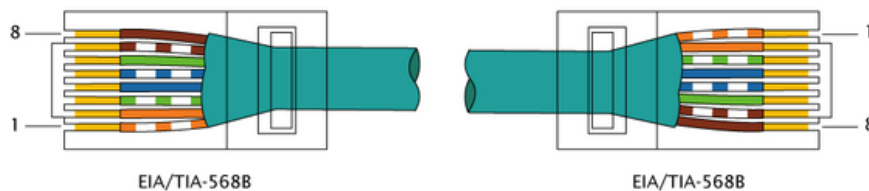


Figura 3.13. Cable directo; Distribución 568B.

No existe diferencia alguna en la conectividad entre la distribución 568B y la distribución 568A siempre y cuando en ambos extremos se use la misma, en caso contrario hablamos de un cable cruzado. El esquema utilizado en este proyecto es la distribución 568B.



Así pues, podríamos haber empleado un cable cruzado para conectar el procesador de comunicaciones CP 343-1 del autómatas con el IRC5, o bien con el PC, sin necesidad de usar el Switch SCALANCE. Sin embargo, éste es necesario ya que se requieren dos puertos Ethernet para comunicación simultánea Autómatas-PC y Autómatas-Controlador de robot.

3.2. Descripción del software utilizado.

3.2.1. Step7 SIMATIC Manager.

El software SIMATIC de Siemens es el entorno de desarrollo mediante el cual hemos creado y configurado la red Ethernet constituida por los tres elementos que componen nuestra aplicación: PC, autómatas y controlador de robot IRC5. También desde este software hemos programado el autómatas.

Concretamente el paquete de software instalado en el PC de trabajo ha sido el **Step7 V5.4 incluido SP3**, que además incluye:

- Paquete .Net Framework 1.1.
- PKWare Inc. Software.
- Automation License Manager V3.

STEP 7 es el software estándar para programar y configurar los sistemas de automatización SIMATIC como nuestro el PLC S7 314C. Dicho software se compone de una serie de aplicaciones o herramientas que por ejemplo nos permitirán:

- ✓ Configurar y parametrizar el hardware.
- ✓ Crear y comprobar los programas de usuario.
- ✓ Configurar segmentos y enlaces.



El software puede ser ampliado con una extensa gama de paquetes opcionales, por ejemplo los paquetes correspondientes a los lenguajes SCL, S7-GGRAPH o HiGraph.

A todas las aplicaciones se accede desde el Administrador SIMATIC, reuniendo en un proyecto todos los datos y ajustes necesarios para su tarea de automatización. Dentro de dicho proyecto, los datos se estructuran por temas y se representan en forma de objetos.

De forma similar a la estructura del Explorador de Windows, compuesta por carpetas y archivos, los proyectos de STEP 7 se subdividen en carpetas y objetos. Las carpetas son aquellos objetos que, a su vez pueden incluir también otras carpetas y objetos, como p.ej. un programa S7 que comprenda las carpetas "Bloques" y "Fuentes", así como el objeto "Símbolos".

Los diversos tipos de objetos están vinculados en el Administrador SIMATIC directamente a la aplicación necesaria para poder editarlos. Gracias a ello, no se necesita recordar qué aplicación deberá arrancar cuando se desee editar un objeto determinado.

Para trabajar con el Administrador SIMATIC, típicamente se siguen estos pasos:

1. Creación de un nuevo proyecto eligiendo para ello el comando Archivo→Nuevo
2. En la ventana del proyecto seleccionamos el símbolo del proyecto.
3. Introducimos un programa S7 con el comando de menú Insertar > Programa > Programa S7.
4. Hacemos doble clic en el programa de usuario (objeto "Bloques") que se encuentre en la carpeta "Programa S7".
5. Hacemos doble clic en el OB1 que aparece en la parte derecha de la ventana del proyecto.

En la aplicación iniciada podremos editar el bloque o crear otros.



6. Determinamos en las propiedades del programa S7 la dirección MPI del módulo en el que desea comprobar los bloques creados. Seleccionamos a tal efecto en el modo de visualización online el programa S7 y elegimos el comando Edición → Propiedades del objeto.
7. Seleccionamos, en el modo de visualización online de la ventana de proyecto, el programa de usuario (carpeta "Bloques") o los bloques que deseamos cargar.
8. Elegimos el comando Sistema de destino → Cargar para cargar en el sistema de destino los objetos seleccionados.

Podemos comprobar el programa de usuario cargado mediante la función “Observar y forzar”.

Ahora nos centraremos en la aplicación que permite crear y probar bloques para las CPUs SIMATIC S7 en los lenguajes de programación KOP, FUP y AWL.

Además de poder crear programas, es decir, de crear y editar bloques lógicos, bloques de datos y tipos de datos de usuario, se ofrecen funciones adicionales para programar, comprobar y poner en marcha el programa:

- ✓ Programar con símbolos.
- ✓ Declarar variables.
- ✓ Leer informaciones de estado y datos de operación de la CPU mediante el comando de menú Información del módulo (Menú Sistema de destino).
- ✓ Visualizar y cambiar el estado operativo de la CPU mediante el comando de menú Estado operativo (Menú Sistema de destino).
- ✓ Borrado total de CPUs.
- ✓ Visualizar y ajustar la fecha y la hora de la CPU mediante el comando de menú Ajustar la hora (Menú Sistema de destino).
- ✓ Crear o visualizar los datos de referencia.



- ✓ Comparar bloques.
- ✓ Comprobar la consistencia de los bloques.
- ✓ Declarar multiinstancias.
- ✓ Generar fuentes AWL.
- ✓ Definir definiciones de errores para el diagnóstico del proceso.
- ✓ Observar un bloque lógico individual.

Para poder crear y editar bloques debe existir un Programa S7. Dicho programa se crea con el Administrador SIMATIC. Si vamos a programar bloques por primera vez, p.ej. con un programa KOP sencillo, es recomendable proceder de la siguiente manera dentro de la aplicación "KOP/FUP/AWL: programar bloques S7":

1. Crear bloques lógicos.
2. Declarar variables.
3. Programar bloques en el área de instrucciones en KOP.
4. Introducir elementos KOP.
5. Editar direcciones y parámetros en KOP
6. Crear ramificaciones paralelas.
7. Crear nuevas ramificaciones.
8. Buscar y eliminar errores.
9. Guardar bloque.
10. Comprobar el programa S7.

3.2.2. Lenguaje RAPID para la programación del robot.

El robot dispone de un mando denominado FlexPendant a través del cual podemos programar el robot en lenguaje RAPID, empleando las instrucciones y creando las variables que sean precisas. FlexPendant es una unidad de programación flexible basada en el único uso de una pantalla gráfica táctil. La presentación general de FlexPendant se apoya en la tecnología de Microsoft, en el sistema operativo Windows CE (sistema operativo de tiempo real para dispositivos móviles).

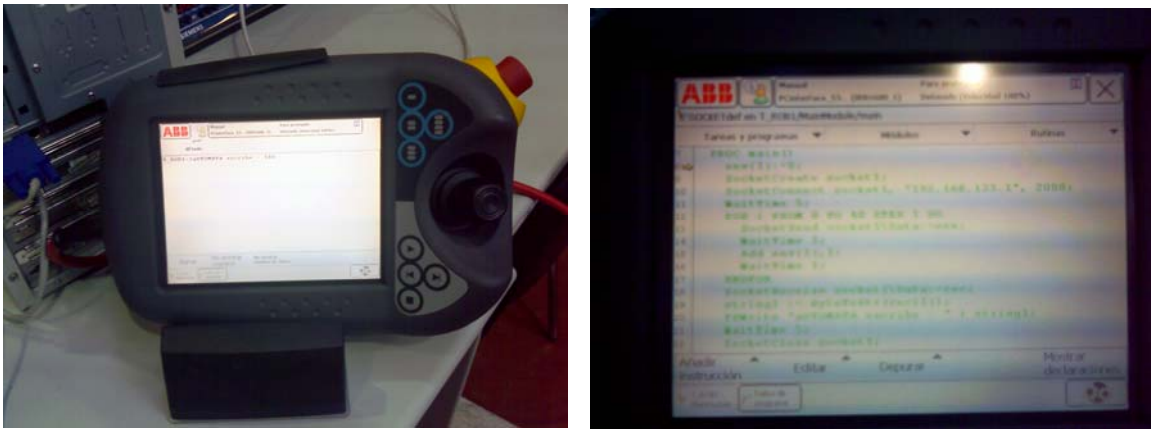


Figura 3.14. FlexPendant del Controlador de robot IRC5.

RAPID (*Robotics Application Programming Interactive Dialogue*) es un lenguaje de programación textual de alto nivel desarrollado por la empresa ABB. Una aplicación RAPID consta de un programa y una serie de módulos del sistema.

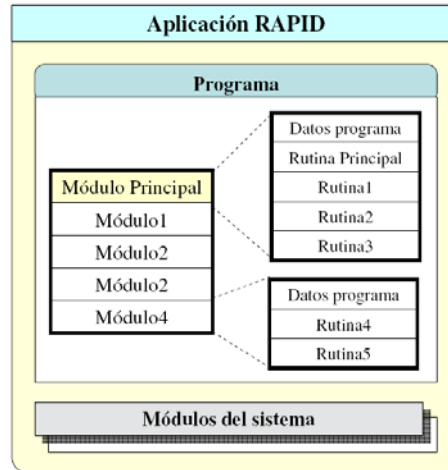


Figura 3.15. Estructura típica de una aplicación RAPID.

El programa es una secuencia de instrucciones que controlan el robot y en general consta de tres partes:

- Una rutina principal (*main*): rutina donde se inicia la ejecución.
- Un conjunto de sub-rutinas: sirven para dividir el programa en partes más pequeñas a fin de obtener un programa modular.
- Los datos de programa: definen posiciones, valores numéricos, sistemas de coordenadas, etc.

El programa de RAPID implementado se muestra y se explica en el punto 4.2.3.2.

3.2.3. Wincc (Windows Control Center) V6.0.1.0.

Se trata de un software mediante el cual hemos creado un ejemplo de aplicación SCADA con el objetivo de probar las comunicaciones. El paquete de software WinCC constituye el entorno de desarrollo de Siemens en el marco de los SCADA para visualización y control de procesos industriales. Sus características más importantes se pueden resumir en:

- ✓ Arquitectura de desarrollo abierta (programación en C).
- ✓ Soporte de tecnologías Active X.
- ✓ Comunicación con otras aplicaciones vía OPC.



- ✓ Comunicación sencilla mediante drivers (código que implementa el protocolo de comunicaciones con un determinado equipo inteligente) implementados.
- ✓ Programación online: no es necesaria detener la runtime del desarrollo para poder actualizar las modificaciones en la misma (la runtime es la ventana que nos permite, en tiempo de ejecución del proceso, interactuar con el mismo).

WinCC es un software IHMI (Integrated Human Machine Interface) que incluye todos los elementos necesarios para controlar y supervisar procesos industriales. Las operaciones principales que podemos desarrollar en WinCC son:

- Edición gráfica de las pantallas.
- Gestión de archivos y bases de datos.
- Generación de informes en base a datos obtenidos.
- Administración de la información.

Típicamente, los pasos para trabajar con un proyecto en WinCC son:

1. Creación del proyecto.
2. Configuración de la conexión con un PLC.
3. Definir variables de comunicación.
4. Creación de las imágenes-pantallas. Para la edición gráfica se ofrecen las opciones típicas (gamas de colores, gama de objetos, asistente dinámico, funciones de alineamiento, funciones de zoom, barra de menús, barra de herramientas, barra de capas, etc.).
5. Definir las características en tiempo de ejecución (RunTime).
6. Programación de variables y objetos.



Por último, comentar que una particularidad de trabajar con WinCC es que el proyecto creado en un determinado PC, sólo puede ser cargado en otro PC distinto si éste último posee el mismo nombre que aquel en el que fue creado.



Capítulo 4:
Desarrollo del Proyecto.

La implementación de las comunicaciones desarrollada en el proyecto podría dividirse en dos fases:

1. Comunicación PC – Autómata (véase punto 4.1).
2. Comunicación Autómata – Controlador de robot (véase punto 4.2).

En el punto 4.3 se integrarán los tres elementos hardware anteriores, explicando qué función desempeña cada uno de ellos en los procesos de envío y recepción de datos de la aplicación desarrollada.

4.1. Comunicación PC – Autómata.

4.1.1. Objetivo, arquitectura y diagrama de flujo.

El objetivo de este punto es establecer una comunicación básica entre el autómata Siemens S7 y el PC.

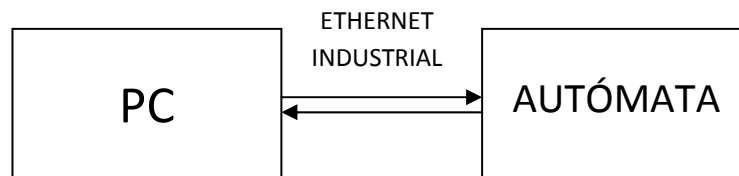


Figura 4.1. Comunicación PC – Autómata.

En el PC, a través de la herramienta software WinCC, activaremos bits o bytes del autómata y también podremos consultar el valor de las posiciones de memoria de éste. Asimismo, desde el autómata podremos, manualmente, efectuar la activación de un bit, que será leído por el PC.

La arquitectura a implementar para esta aplicación sería la siguiente:

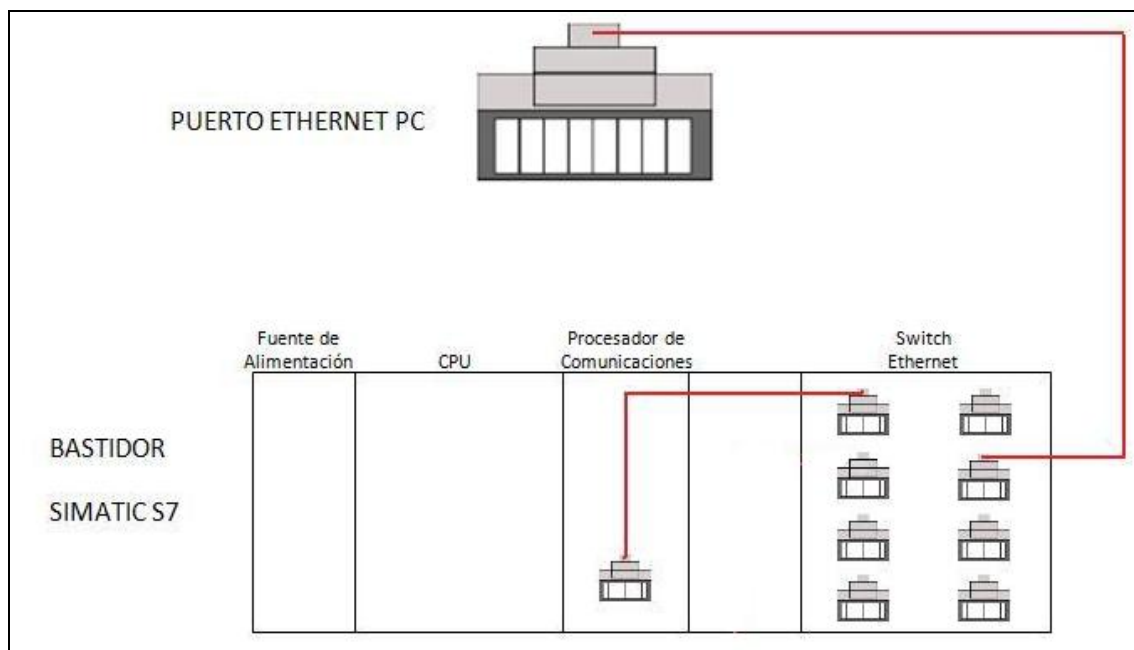


Figura 4.2. Arquitectura de comunicación PC – Autómata.

Para entender mejor el funcionamiento de este subapartado se muestra el siguiente diagrama de flujo.

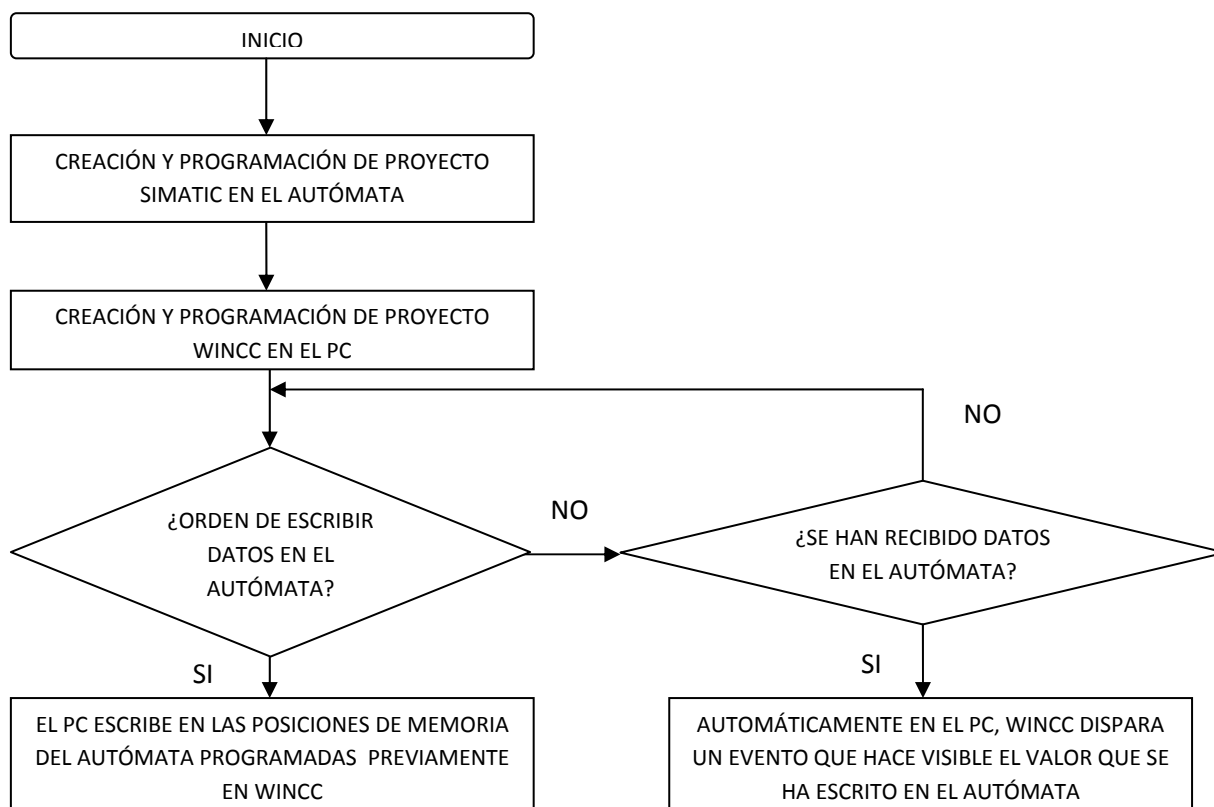


Figura 4.3. Diagrama de flujo de la comunicación PC – Autómata.

En primer lugar crearemos un proyecto SIMATIC Manager mediante el cual se configura el hardware del autómata y se crea la red Ethernet Industrial entre el PC y el autómata.

En el PC, con WinCC se ha desarrollado una simulación de SCADA (*Supervisory Control and Data Acquisition*) para probar las comunicaciones de la aplicación implementada. Las pantallas creadas se muestran en el punto 4.3.3. del presente proyecto. Para el envío de datos desde el SCADA, bastará con hacer click sobre un botón creado; la recepción de los datos enviados por el controlador de robot (que previamente pasarán por el autómata) será visible al cambiar de estado una imagen del robot que describe la tarea ejecutada por éste en cada instante.

Así pues, la orden de activar bits en el autómata se efectuará haciendo clic sobre un botón creado en WinCC. Por otro lado, en el autómata se podrán recibir datos, aparte de los procedentes del PC, desde el controlador de robot o manualmente mediante los interruptores del mismo.

4.1.2. Configuración y programación en Wincc. Creación de variables.

Lo primero que tenemos que hacer es crear un proyecto nuevo. Para ello vamos a Archivo→Nuevo y marcamos la opción “Proyecto para estación monopuesto”, ya que en nuestra aplicación sólo tenemos un autómata.

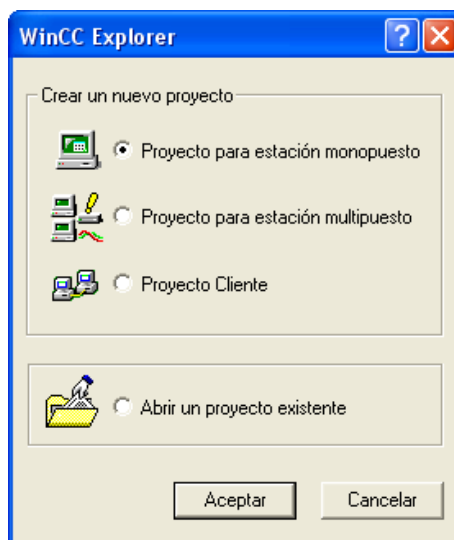


Figura 4.4. Creación de un proyecto monopuesto en WinCC.



A continuación creamos los canales de comunicación entre el pc y el autómata. Nos situamos en “Administración de Variables” en el explorador de WinCC:



Figura 4.5. Administrador de variables en WinCC.

Pulsamos el botón derecho y le damos a “Agregar nuevo driver”, y nos aparecerá la siguiente ventana donde abriremos el driver “SIMATIC S7 Protocol Suite”:

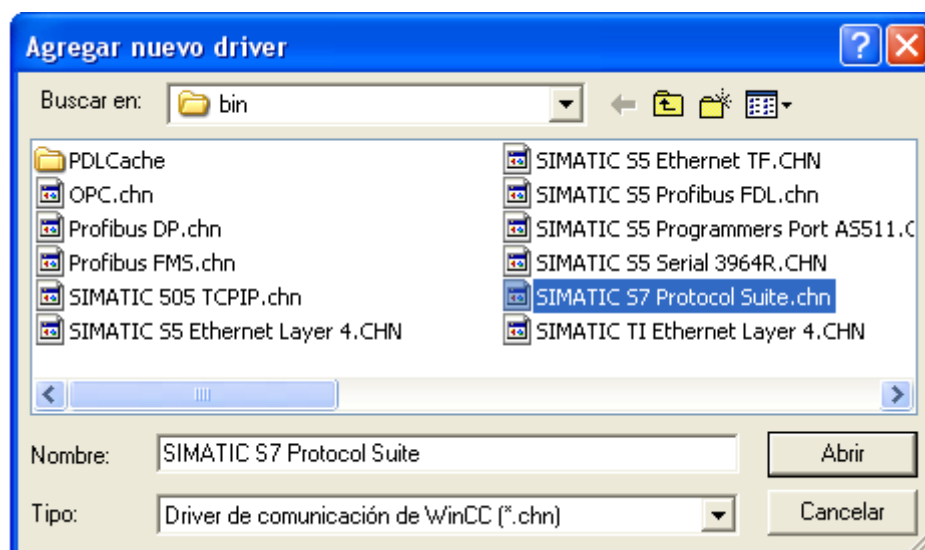


Figura 4.6. Agregar driver de comunicación de WinCC.

Una vez hecho esto, en WinCC podremos observar todas sus posibles interfaces de comunicación con el exterior: MPI, PROFIBUS, Industrial Ethernet, Slot PLC y TCP/IP.

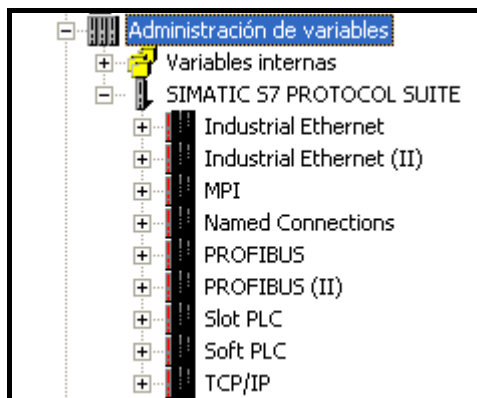


Figura 4.7. Interfaces de comunicación del driver Simatic S7 Protocol Suite.

Como hemos explicado en anteriores puntos, la comunicación entre el PC y el autómata se realizará por Industrial Ethernet ya que es la red más eficaz en lo que se refiere a rapidez y costes (aunque también se ha utilizado como segunda opción la comunicación por MPI, siendo la configuración en WinCC muy similar al proceso seguido para Industrial Ethernet que se describe a continuación).

En el campo “Industrial Ethernet” creamos una nueva conexión, que en este caso se ha llamado ComEthernet.

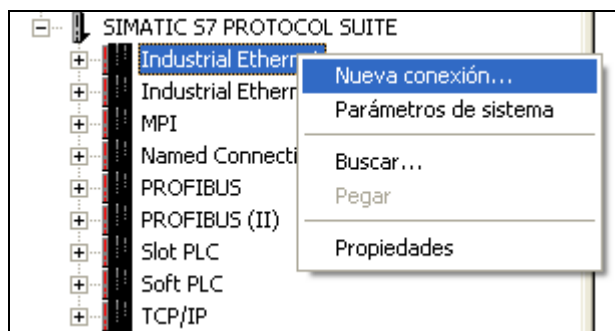


Figura 4.8. Creación de una nueva conexión por Industrial Ethernet.

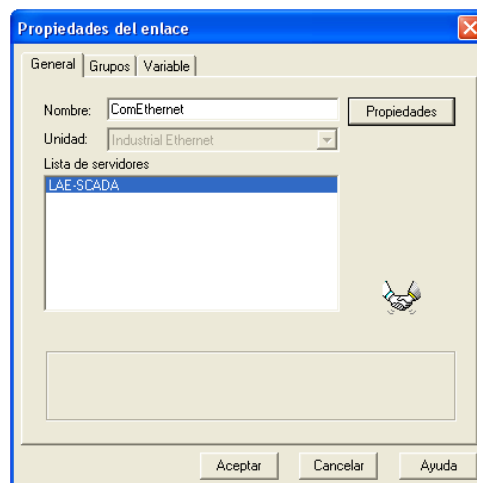


Figura 4.9. Propiedades de la nueva conexión Ethernet creada.

En propiedades deberemos indicar la dirección Ethernet (MAC) del autómata, así como el número de slot y bastidor de la CPU.

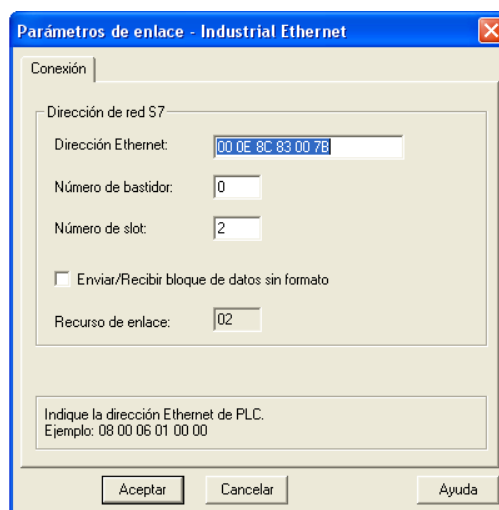


Figura 4.10. Parámetros de la conexión por Industrial Ethernet.

Con estos pasos ya tendríamos la conexión Industrial Ethernet disponible para su programación.

El elemento clave en la programación con WinCC son las variables. Mediante ellas podremos, desde el pc, leer y escribir datos en la CPU del autómata. Para crear una variable nos iremos a la conexión Industrial Ethernet creada anteriormente, pulsaremos el botón derecho y seleccionaremos “Variable nueva”.

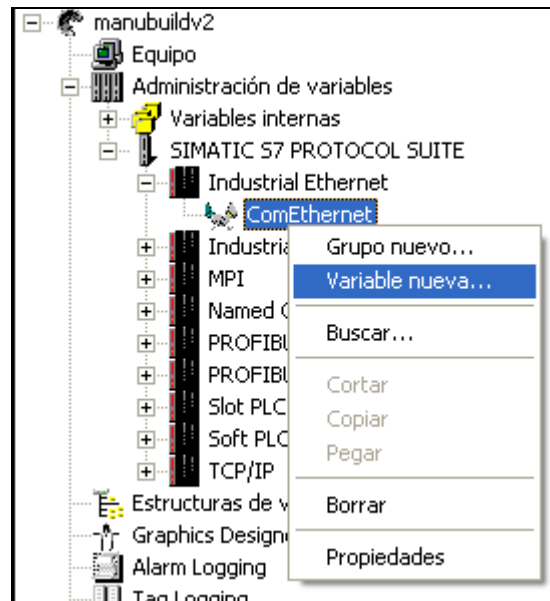
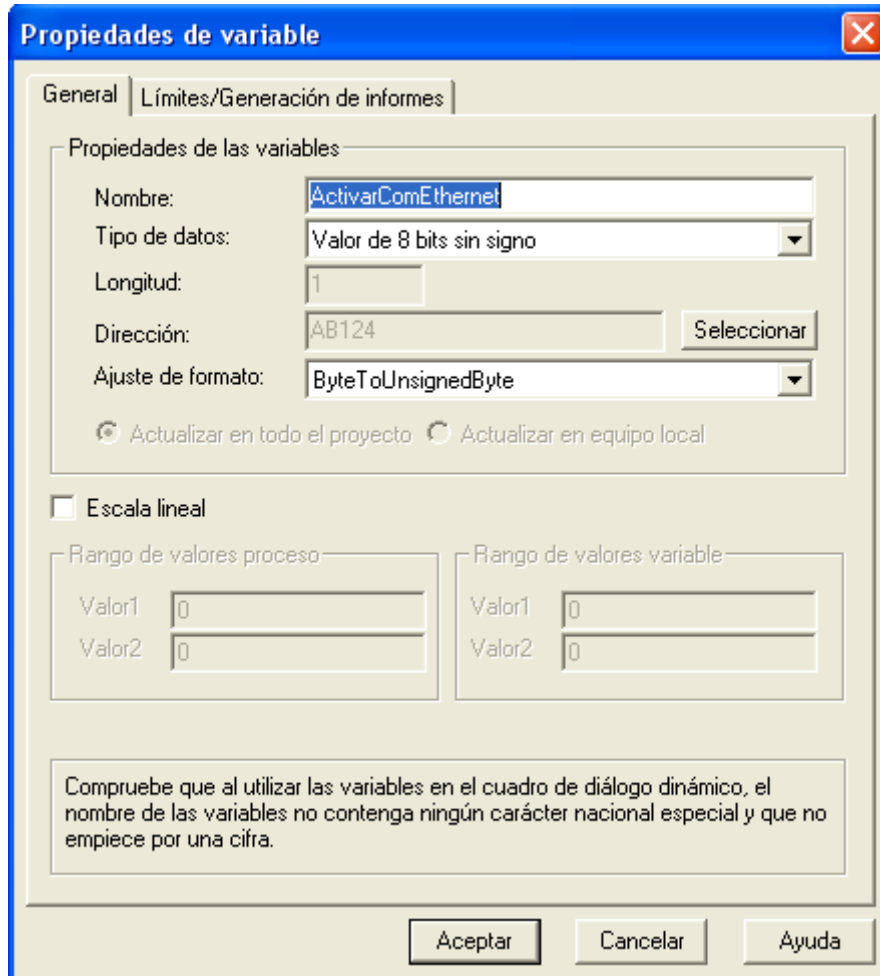


Figura 4.11. Creación de variables en Industrial Ethernet.

Nos aparecerá una ventana de propiedades, donde daremos un nombre a la variable y realizaremos dos pasos importantes:

- Seleccionamos el tipo de datos de la variable, que puede ser variable binaria (bit), un byte, una palabra de 16 bits o de 32 bits, con o sin signo.
- En el campo “Dirección”, seleccionamos el área de datos donde se localizará la variable dentro de la memoria del PLC (entrada, salida, DB o marcas) así como su dirección inicial.



Propiedades de variable

General | Límites/Generación de informes

Propiedades de las variables

Nombre:

Tipo de datos:

Longitud:

Dirección:

Ajuste de formato:

☒ Actualizar en todo el proyecto ☐ Actualizar en equipo local

☐ Escala lineal

Rango de valores proceso

Valor1

Valor2

Rango de valores variable

Valor1

Valor2

Compruebe que al utilizar las variables en el cuadro de diálogo dinámico, el nombre de las variables no contenga ningún carácter nacional especial y que no empiece por una cifra.

Figura 4.12. Establecimiento de las propiedades de la variable Ethernet creada.

Si queremos escribir en el PLC la variable será de salida (A) y si queremos leer del PLC la variable será de entrada (E). Los bloques de datos (DB) y marcas pueden ser tanto de entrada como de salida.



Figura 4.13. Direccionamiento de la variable.

Para la comunicación entre el PC y el autómata por Industrial Ethernet se han definido dos variables:

- ActivarComEthernet

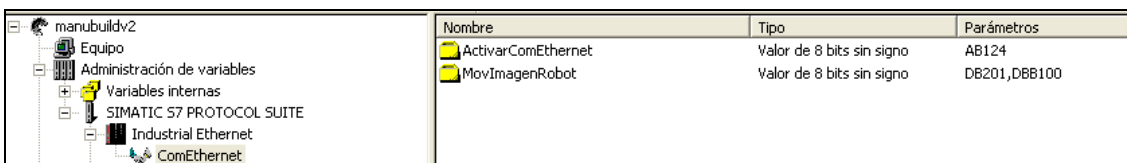
Variable de salida de tipo byte en la dirección 124.

Mediante esta variable activaremos un bit del PLC al pulsar un botón (ver figura 4.94 del apartado 4.3.3. “Ejemplo de aplicación SCADA”), lo que a su vez hará que se inicie el envío de datos vía ethernet desde el autómata al controlador del robot.

- MovImagenRobot

Se trata de un bloque de datos (DB201) de tipo byte.

En el byte 100 (DBB100) el autómata recibirá los datos enviados por el robot. WinCC irá leyendo estos datos, y en función de ellos, mostrará una imagen del robot descriptiva de la tarea ejecutada por éste, de tal forma que el usuario del PC sabrá en todo momento cuál es la función que está realizando el robot (Véanse las tablas 4.1 y 4.2 del punto 4.3.3).



| Nombre | Tipo | Parámetros |
|--------------------|---------------------------|--------------|
| ActivarComEthernet | Valor de 8 bits sin signo | AB124 |
| MovImagenRobot | Valor de 8 bits sin signo | DB201,DBB100 |

Figura 4.14. Variables utilizadas en la comunicación por Industrial Ethernet.

Hasta aquí se ha explicado lo referente a la comunicación por Ethernet Industrial con WinCC. A continuación, se explicará lo relacionado con la interfaz MPI, que también ha sido utilizada en este proyecto.

La interfaz MPI es utilizada únicamente para cargar los programas al autómata. Es la interfaz Ethernet Industrial la que se emplea para el proceso de comunicación en sí mismo, es decir, el tráfico de información entre el PC, el autómata y el controlador de robot se realiza a través de Ethernet Industrial y no a través de la interfaz MPI. No obstante, la comunicación PC – Autómata puede efectuarse también por MPI. Así pues, en cuanto a la programación en WinCC, todo el proceso explicado hasta ahora para Ethernet Industrial podría realizarse de igual forma para la interfaz MPI. A continuación se expone dicho proceso.

Nos situaremos sobre el campo MPI y haremos clic en “Nueva Conexión”, que hemos llamado ComMPI.

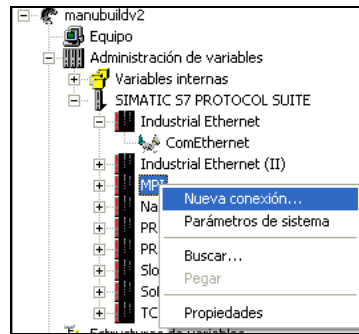


Figura 4.15. Creación de una nueva conexión MPI.

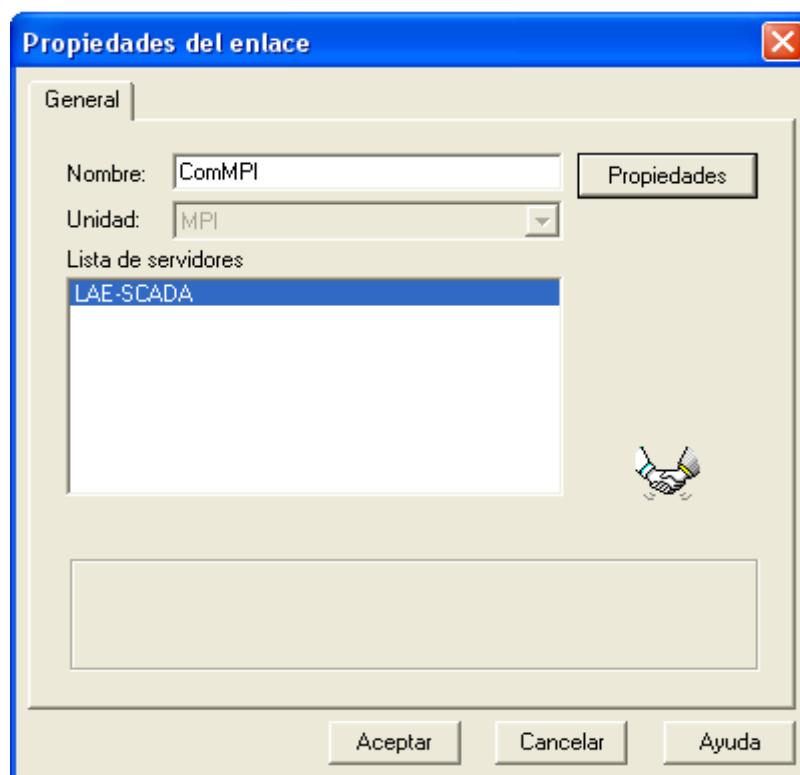


Figura 4.16. Propiedades de la conexión MPI.

En “Propiedades” configuraremos los parámetros del enlace (la dirección de la estación del PLC y el número de bastidor y slot de la CPU del autómata).

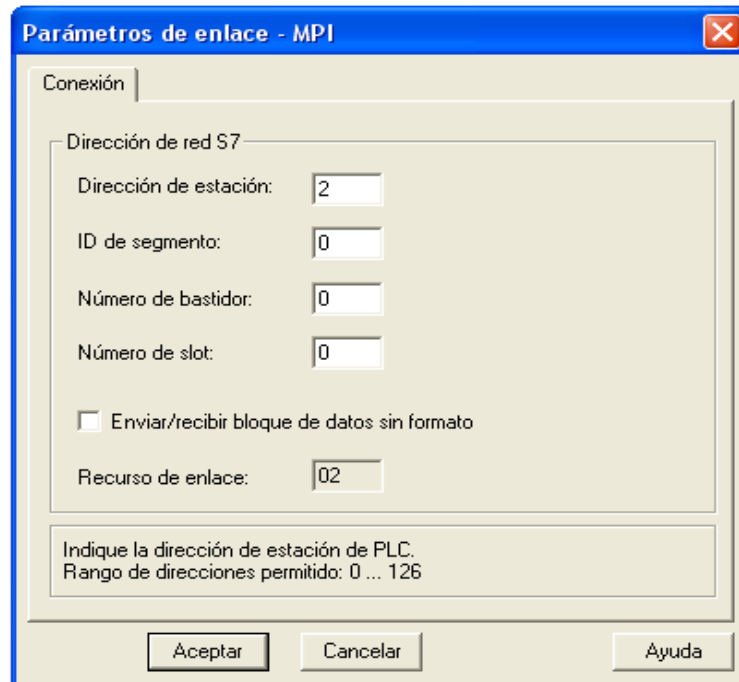


Figura 4.17. Parametrización del enlace MPI.

Con estos pasos ya tendríamos la conexión MPI disponible para su programación. Las variables creadas para esta conexión han sido dos:

| | Nombre | Tipo | Parámetros |
|-----------------------------|---------------------|---------------------------|------------|
| manubuildv2 | Estructura/Baldosas | Valor de 8 bits sin signo | AB125 |
| Equipo | DatosRecibidosMPI | Variable binaria | E124.0 |
| Administración de variables | | | |
| Variables internas | | | |
| Script | | | |
| TagLoggingRt | | | |
| SIMATIC S7 PROTOCOL SUITE | | | |
| MPI | | | |
| ComMPI | | | |
| PROFIBUS | | | |
| Industrial Ethernet | | | |

Figura 4.18. Variables de la conexión MPI.

- Estructura/Baldosas

Se trata de una variable de salida tipo byte de dirección 125, que usaremos en la pantalla 2 (ver figura 4.93 del apartado 4.3.3. “Ejemplo de aplicación SCADA”) para indicarle al robot qué demostración queremos efectuar: el ensamblaje de una estructura o cortar baldosas.

- DatosRecibidosMPI

Es un bit de entrada cuya dirección es 124.0. Será utilizado en la pantalla 4 para indicar que una persona ha pulsado el interruptor del autómata relativo a dicha dirección.

El proceso para crear estas variables es análogo al explicado para la interfaz Ethernet Industrial. Haciendo clic derecho sobre el nombre de la conexión MPI, pulsamos en “Variable nueva” y nos aparecerá la siguiente ventana.

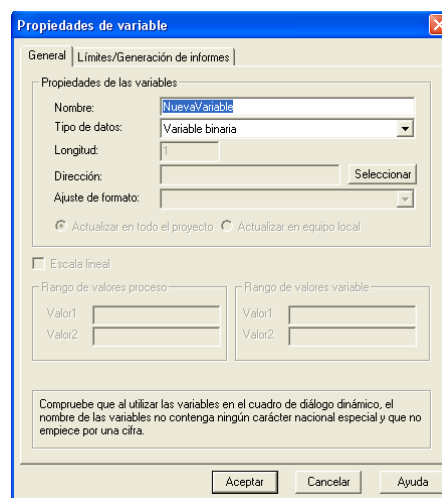


Figura 4.19. Propiedades de la variable MPI.

Aquí daremos un nombre a la variable, seleccionaremos si queremos que sea de tipo bit, byte, 16 o 32 bits, y también su posición (dirección) en la memoria del PLC, pudiendo seleccionar entre el área de entradas, el área de salidas, el área de marcas y el área de bloques de datos.

4.1.3. Creación de pantallas y uso de variables en WinCC

Nos queda saber cómo crear las pantallas del SCADA y cómo programar en ellas las variables anteriores.

Nos situaremos en el campo “Graphics Designer” del explorador de Wincc, pulsaremos el botón derecho, seleccionaremos “Nueva Imagen” y le daremos un nombre. A través

de esta herramienta crearemos las pantallas utilizando las opciones típicas de edición gráfica que se nos ofrecen: insertar objetos, formas, botones, textos, ajustar propiedades, colores, etc.

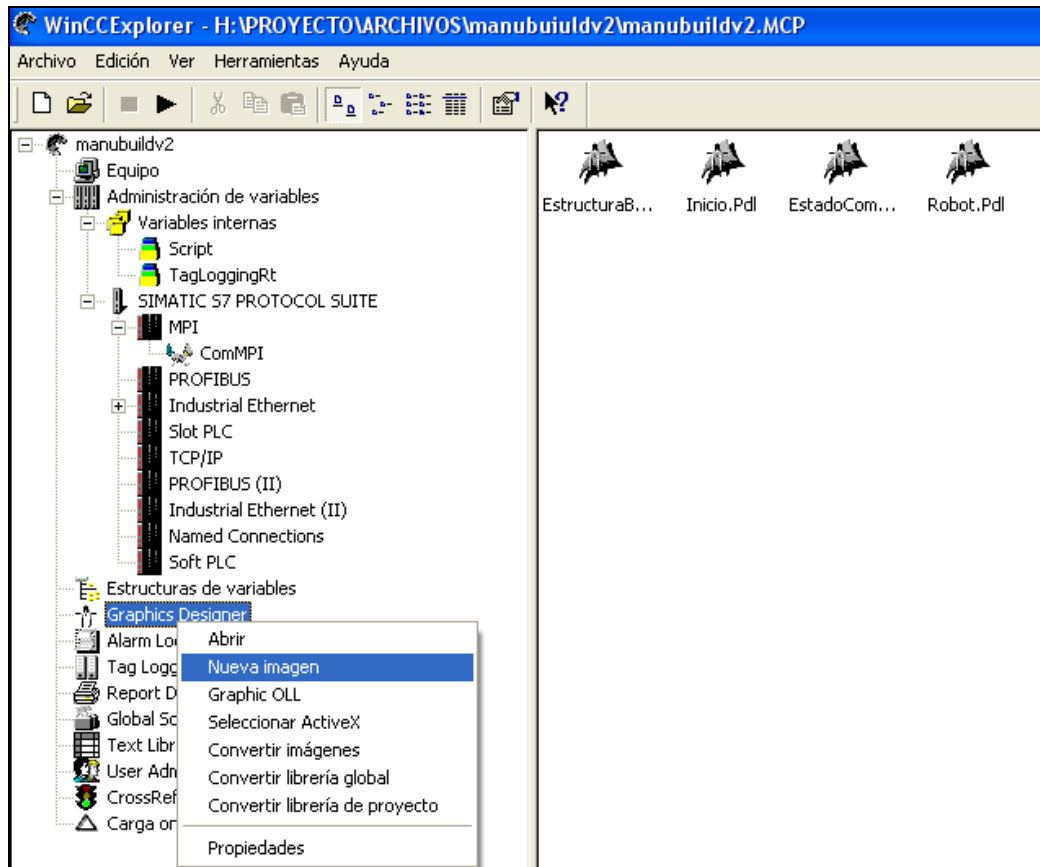


Figura 4.20. Abrir el editor de pantallas Graphics Designer.

Como puede observarse, en la simulación de SCADA desarrollada para el presente proyecto se han creado cuatro pantallas. A continuación se muestran y se explica el uso de las variables en éstas.

1. Inicio.Pdl



Figura 4.21. Pantalla de inicio del SCADA.

La pantalla de inicio muestra el logotipo de Manubuild, un proyecto sobre la aplicación de la Robótica y la Automatización en el campo de la Construcción.

Para crear el botón “Acceder” nos vamos a “Objetos Windows” de la Paleta de Objetos.

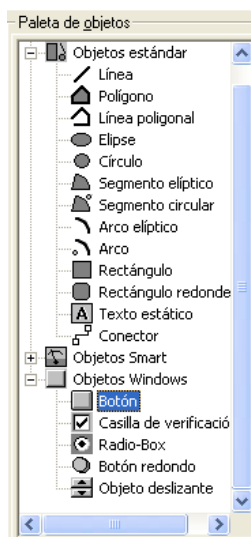


Figura 4.22. Paleta de objetos del Graphics Designer.

Mediante el botón derecho accederemos a su configuración y le indicaremos que al hacer clic sobre él, salte a la pantalla siguiente EstructuraBaldosas.Pdl.

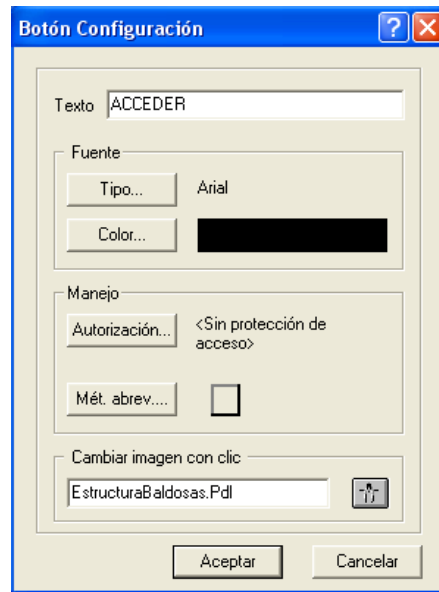


Figura 4.23. Configuración del botón ACCEDER.

Para crear el botón “Salir” se ha utilizado el Asistente Dinámico que nos ofrece la función de sistema “Salir de WinCC Runtime”. Llamamos Runtime a la ventana (formada por diversas pantallas) desde la cual se controla un proceso en tiempo real, recibiendo y enviando información al mismo. Es decir, es la interfaz proporcionada al usuario del pc para comunicarse con el proceso que queremos controlar.

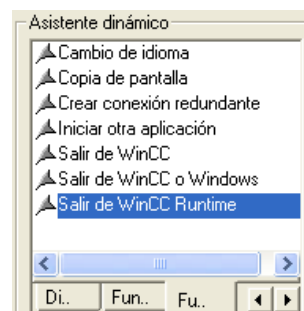


Figura 4.24. Asistente para salir de Runtime.

Para fijar esta imagen como la inicial, lo haremos en el explorador de WinCC en Equipo->Propiedades->Propiedades->Runtime de Graphics, donde también podremos elegir los atributos de ventana. Es importante que en la pestaña “Arranque” estén



marcadas las opciones “Global Script Runtime” y “Graphics Runtime”, para permitir sin problemas la ejecución de las acciones programadas. Hay que tener en cuenta que detrás de la programación mediante menús que se está explicando, WinCC introduce por sí mismo código C que es lo que verdaderamente se ejecuta al tener lugar un evento.

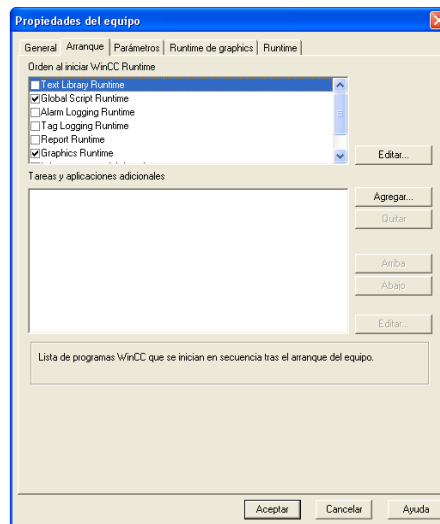


Figura 4.25. Ajuste de programas ejecutados al iniciar la Runtime.

2. EstructuraBaldosas.Pdl

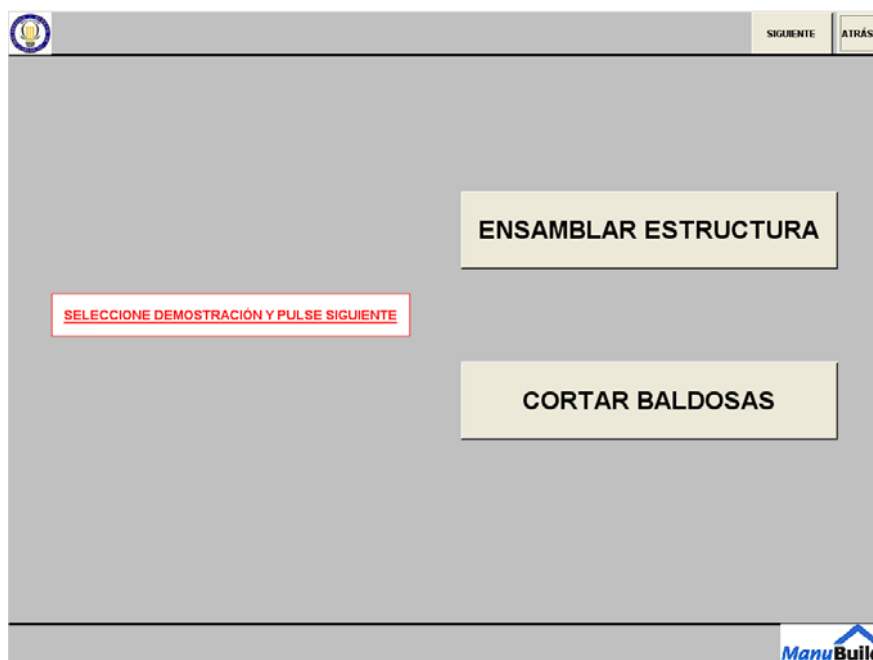


Figura 4.26. Pantalla de selección de operación.

En esta pantalla, el usuario elegirá qué operación desea llevar a cabo con el robot, ensamblar una estructura o cortar baldosas. Para ello utilizaremos la variable de salida tipo byte Estructura/Baldosas ya declarada anteriormente. Activaremos en el PLC el bit 125.1 si queremos ensamblar estructura, y el bit 125.2 si queremos cortar baldosas. Esto lo haremos de la siguiente forma:

1) Desde el Graphics Designer, seleccionamos el botón y abrimos el asistente dinámico Activar/Desactivar bit.

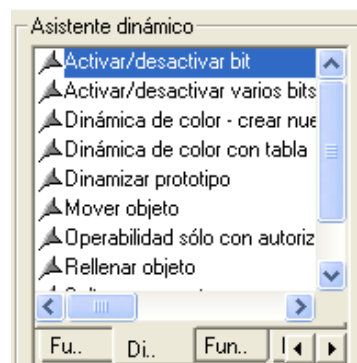


Figura 4.27. Asistente dinámico para activar/desactivar bit.

2) Seleccionamos cuando queremos que se produzca el evento (la activación del bit).

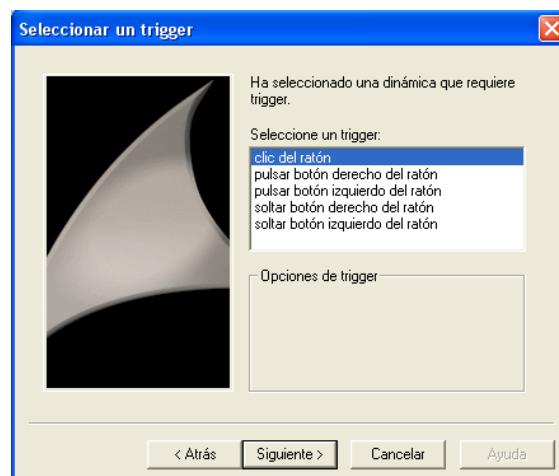


Figura 4.28. Selección del momento de activación del evento.

3) Seleccionamos la variable y la acción a ejecutar.

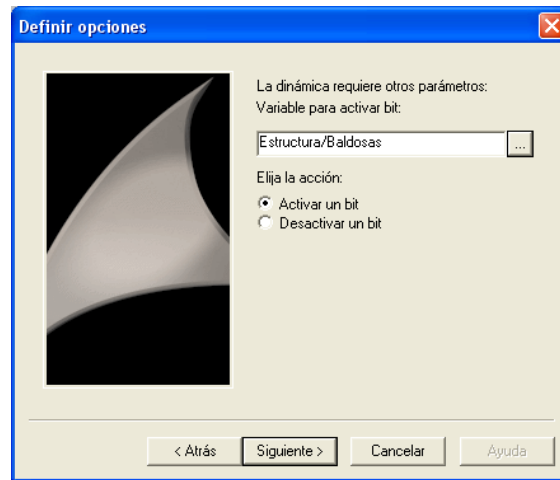


Figura 4.29. Selección de la acción a ejecutar tras evento.

4) Seleccionamos el bit, en este caso el 1.

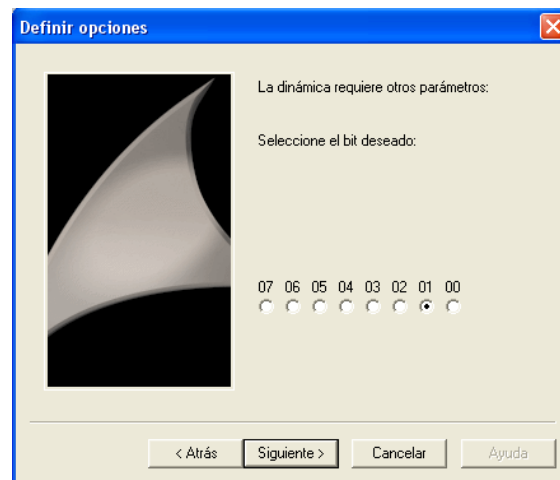


Figura 4.30. Selección del bit.

El ejemplo anterior corresponde a la activación del bit del PLC al pulsar el botón “Ensamblar Estructura”. Para el bit de “Cortar Baldosas”, únicamente cambiaría el paso 4 para seleccionar el bit 2.

Hasta ahora lo que se ha hecho es escribir en el PLC. Ahora, con la siguiente pantalla, vamos a ver un ejemplo de cómo leer del PLC.

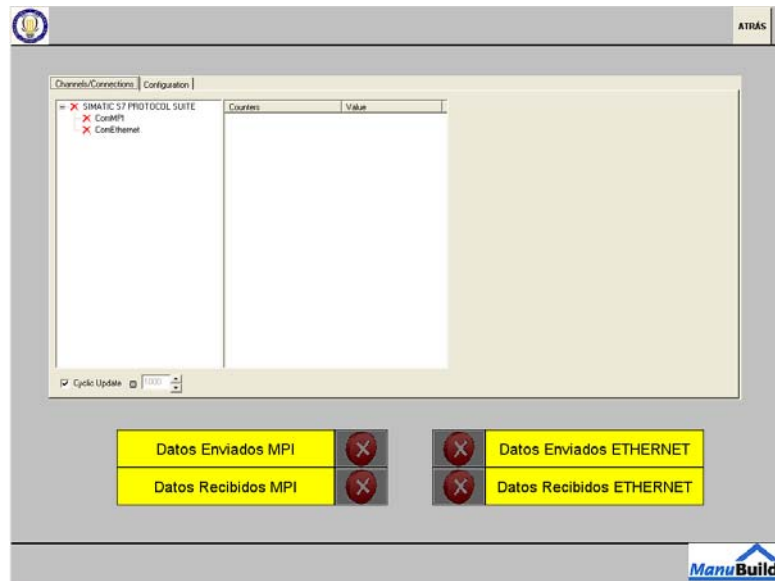


Figura 4.31. Pantalla de diagnóstico de las comunicaciones.

Lo que queremos hacer es que cuando una persona active manualmente, mediante un interruptor, un bit en el autómata, la casilla “Datos recibidos por MPI” pase del estado que vemos en pantalla al estado OK. Esto lo haremos empleando la variable de entrada binaria creada anteriormente “DatosRecibidosMPI” cuya dirección es 124.0

En la paleta de objetos, insertaremos el objeto smart “Vista de estado”, el cual chequea el valor de una variable dada, y en función del valor de ésta, muestra una imagen.

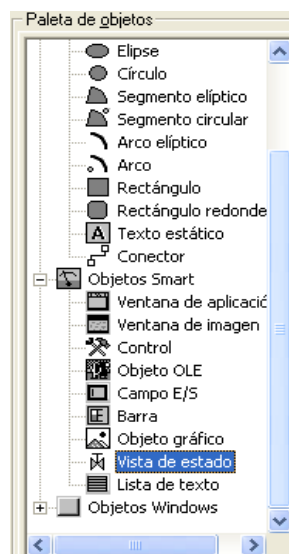


Figura 4.32. Selección del objeto Vista de estado.

Si E124.0=0 → No he activado el bit → La imagen de estado es una cruz.

Si E124.0=1 → He activado el bit → La imagen de estado es un tick.

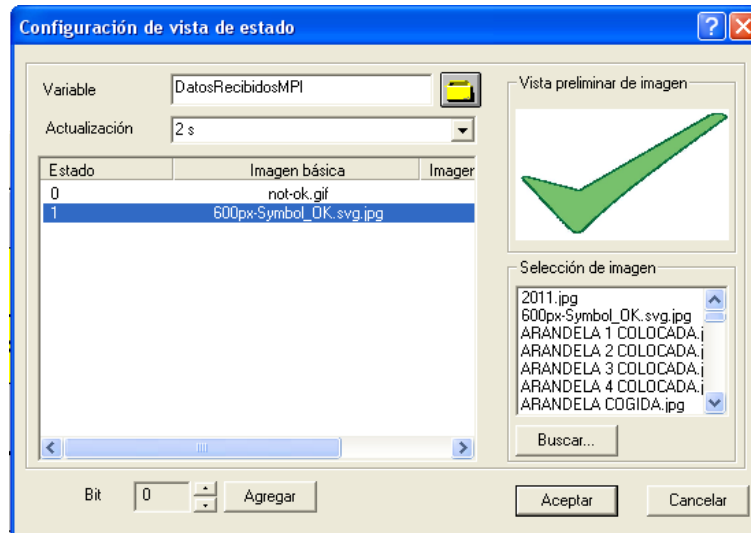


Figura 4.33. Configuración de imágenes del objeto Vista de estado.

Otra forma alternativa de diagnosticar el estado de las comunicaciones en WinCC puede hacerse desde Herramientas→Estado de las conexiones, una vez que se ha activado el Runtime.

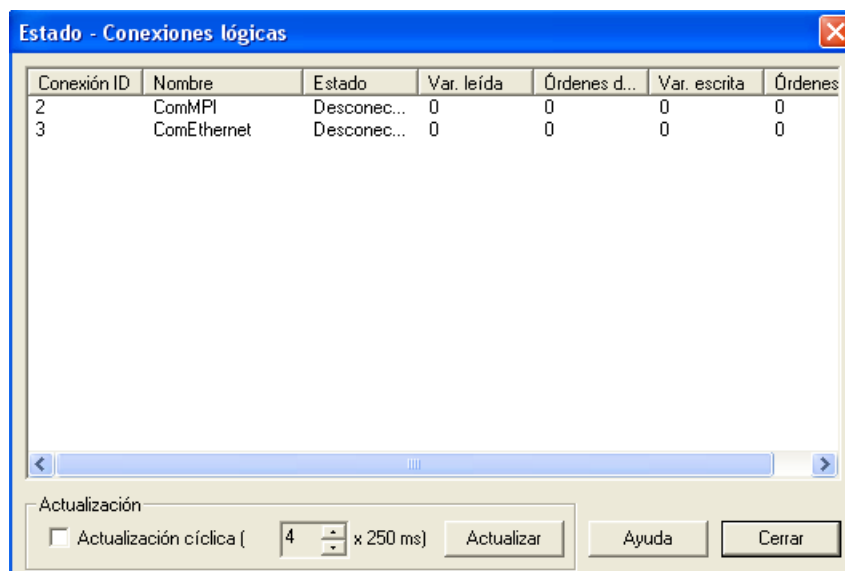


Figura 4.34. Visualización del estado de las comunicaciones.

Por último, en la siguiente pantalla se han usado las dos variables pertenecientes a la conexión Ethernet.

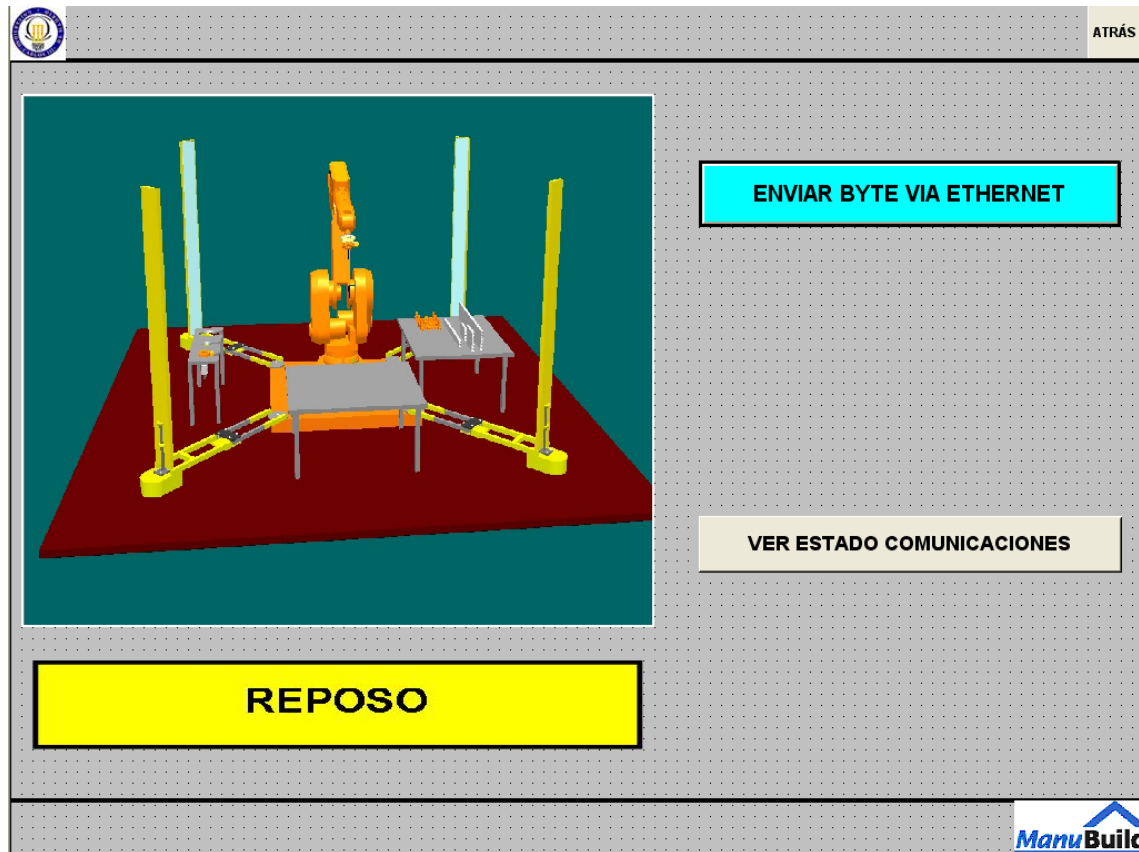


Figura 4.35. Pantalla de manejo de variables Ethernet.

Para representar una imagen y un texto descriptivo de la tarea ejecutada por el robot en cada instante, se han empleado dos objetos “Vista de Estado”, que irán cambiando en función del byte que envía el IRC5. Dicho byte transmitido desde el controlador de robot, se almacena en la variable “MovImagenRobot”, que como puede verse en el apartado 4.1.2 de creación de variables en Industrial Ethernet, es el byte 100 del DB201. Este buffer de recepción es tratado más adelante en el punto 4.2.3.1. “Programación del autómata en Step7”. No obstante como adelanto se muestra el bloque utilizado para recibir datos en el autómata, donde en el campo RECV podemos ver la dirección de recepción mencionada DB201.DBB100.

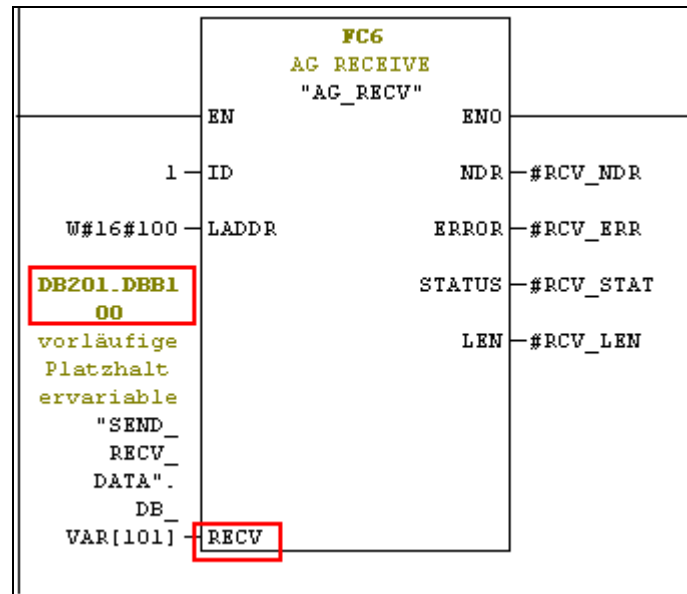


Figura 4.36. Bloque de recepción de datos en el autómata.

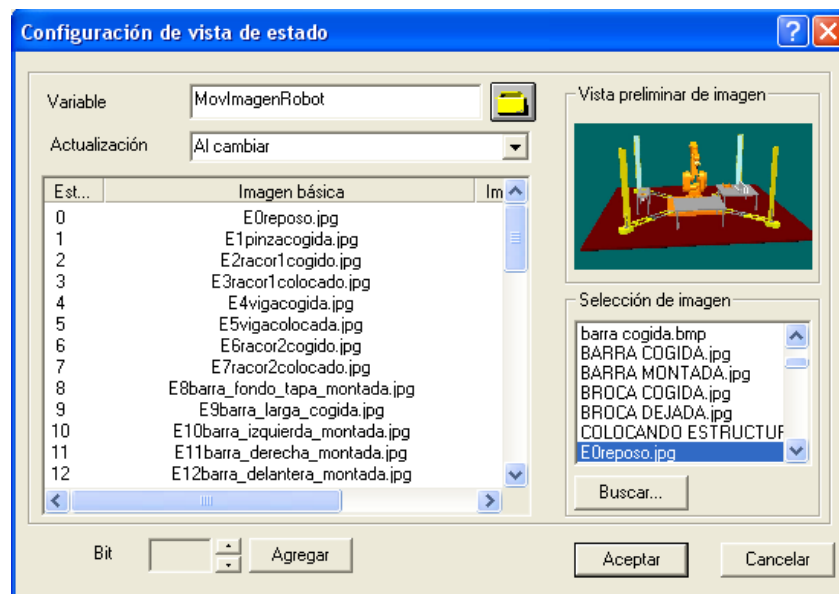


Figura 4.37. Animación de imágenes de la tarea ejecutada por el robot.

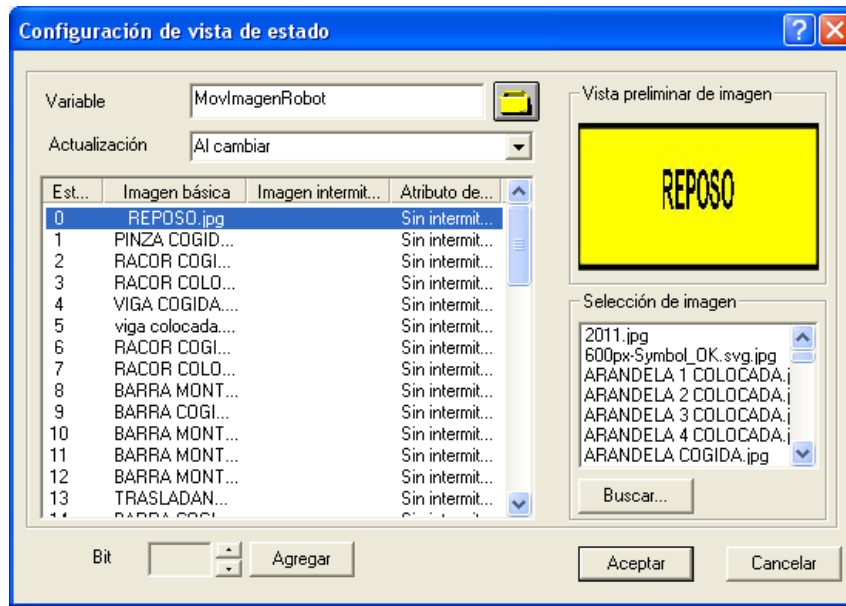


Figura 4.38. Configuración de textos descriptivos de la tarea ejecutada por el robot.

Los diferentes estados por los que pasa el proceso, un total de 49 en la aplicación desarrollada, pueden verse en el punto 4.3.3. “Ejemplo de aplicación SCADA”.

Por otro lado en el botón “Enviar byte Via Ethernet” hemos usado la variable ActivarComEthernet. Al pulsar este botón activaremos el bit A124.1, lo cual provocará que se inicie el envío de datos. En la figura siguiente vemos que el bit A124.1 es el valor del campo ACT (Bit de Activación de envío).

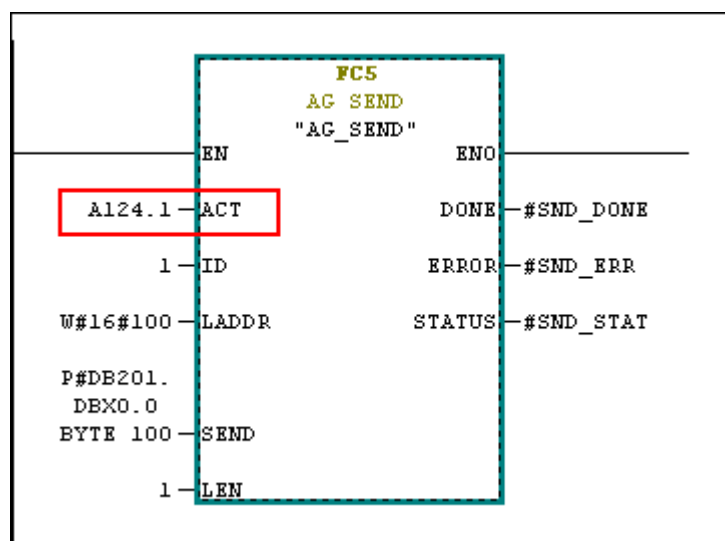


Figura 4.39. Bloque para el envío de datos desde el autómata.

4.1.4. Configuración y programación en Simatic Manager. Creación de la red Ethernet.

Para empezar a crear la red Ethernet, primero nos centraremos en la conexión PC-autómata y una vez hecho esto, se incluirá el controlador de robot a la red.

Desde el SIMATIC Manager, lo primero que tenemos que hacer es ajustar la interfaz a través de la cual nuestro pc se comunicará con el exterior, es decir, la tarjeta de red. Para ello nos iremos a Herramientas → Ajustar interfaz PG/PC y seleccionaremos la tarjeta de red Ethernet de que disponga nuestro pc, en este caso una NIC Fast Ethernet PCI Familia RTL8139 de Realtek. El protocolo utilizado será el ISO Industrial Ethernet.

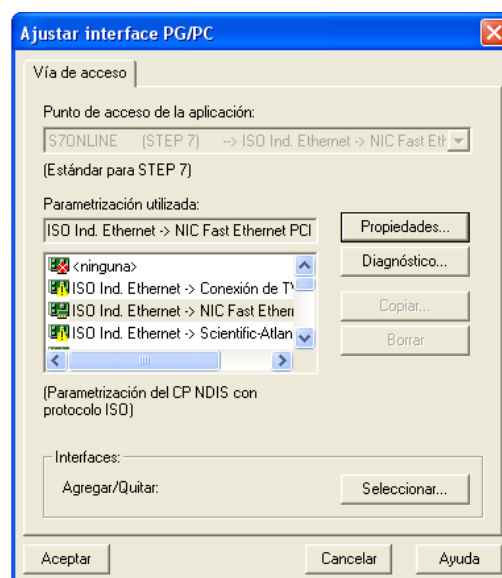


Figura 4.40. Selección de la tarjeta Ethernet del PC como interfaz.

A continuación nos crearemos un nuevo proyecto, e insertaremos dos de los tres equipos que componen la red: PC y autómata.

Para insertar el autómata nos iremos al nombre del proyecto → Insertar nuevo objeto → SIMATIC 300.

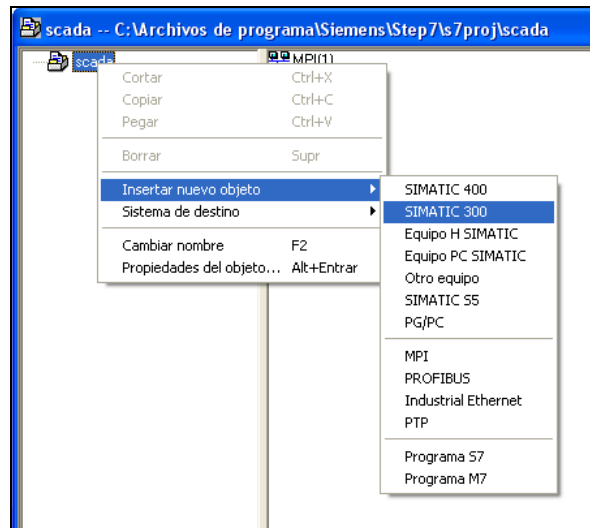


Figura 4.41. Insertar el autómata SIMATIC 300.

Ahora haremos doble clic en AUTÓMATA→Hardware y se nos abrirá la herramienta HWConfig, donde seleccionaremos los módulos que lo componen.

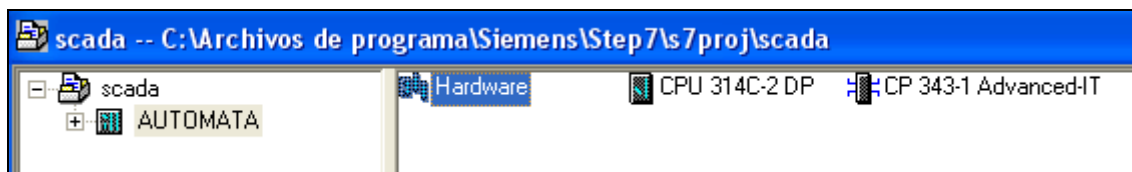


Figura 4.42. Configuración hardware del autómata.

Primero insertaremos el bastidor del autómata: Insertar objeto→SIMATIC 300→BASTIDOR 300→ Perfil Soporte

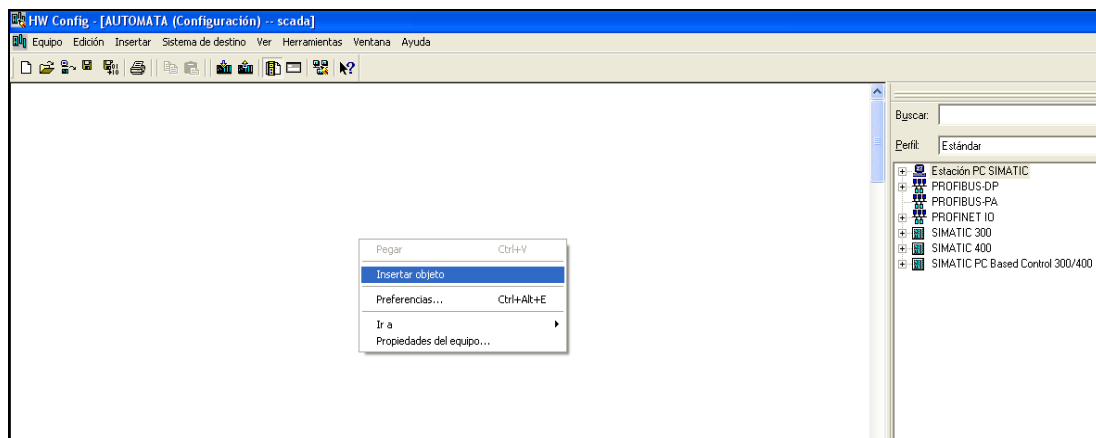


Figura 4.43. Insertar bastidor del autómata 1/2.

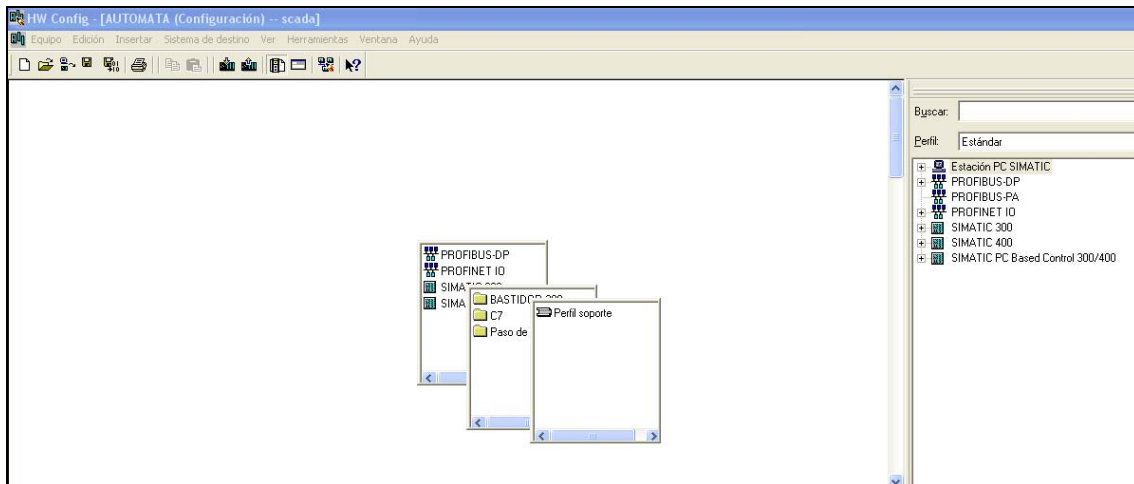


Figura 4.44. Insertar bastidor del autómata 2/2.

Ahora tenemos que ir insertando los distintos módulos que forman el autómata. Para ello los seleccionaremos en la columna de la derecha y los arrastraremos al slot en el que se encuentran.

Slot 1. Fuente de alimentación PS 307 2A

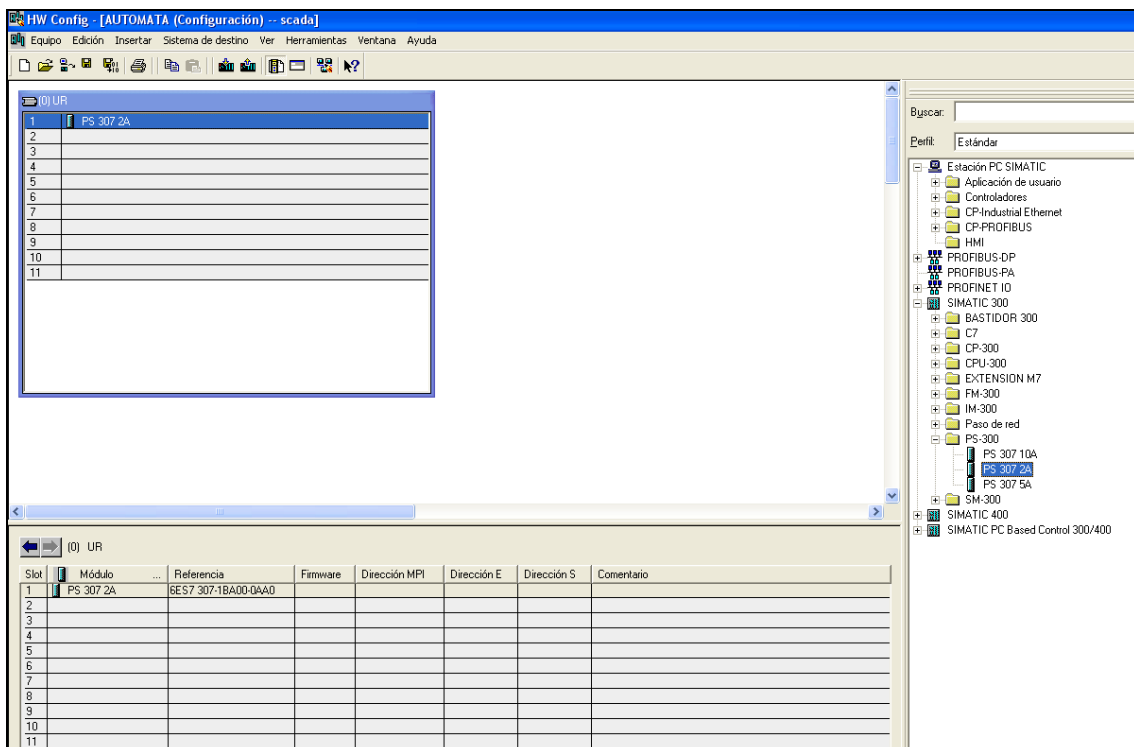


Figura 4.45. Configuración hardware del autómata, Fuente de Alimentación.



Slot 2. CPU 314C-2 DP 6ES7 314-6CG03-0AB0 / v2.0.12

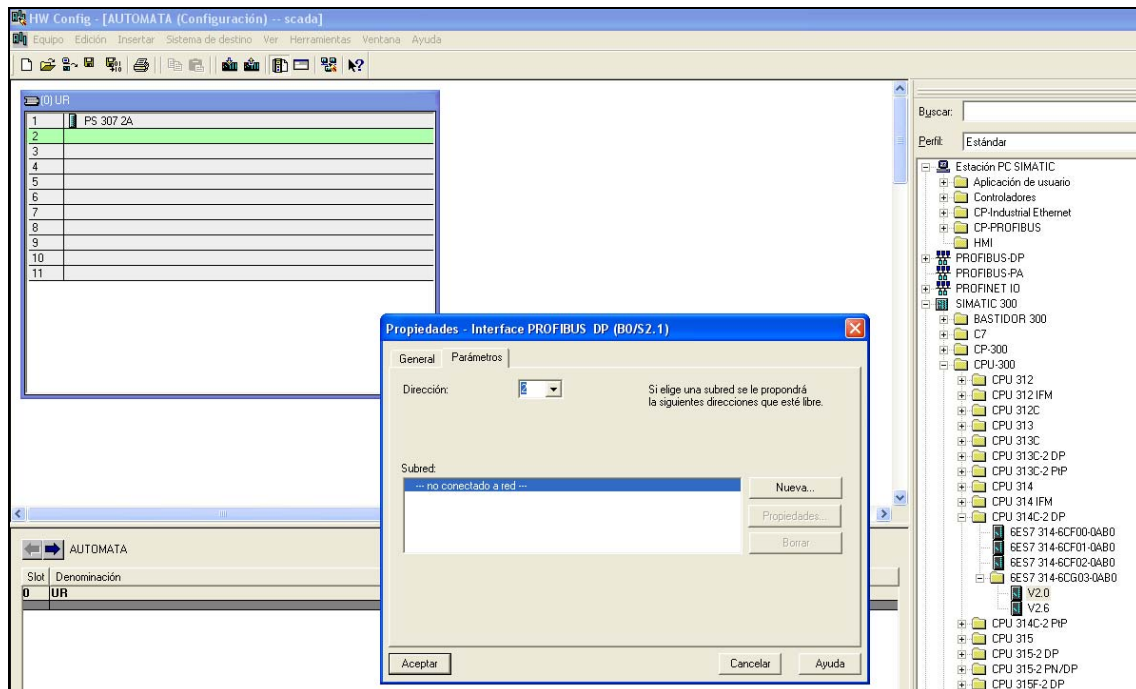


Figura 4.46. Configuración hardware del autómata, CPU.

Al insertar la CPU se nos ofrece la posibilidad de crear una red PROFIBUS y asociar la CPU a dicha red. Cancelaremos esta opción ya que en esta práctica la interfaz de comunicación elegida es Industrial Ethernet.



Slot 4. CP 343-1 ADV-IT 6GK7 343-1GX21-0XE0 V1.1

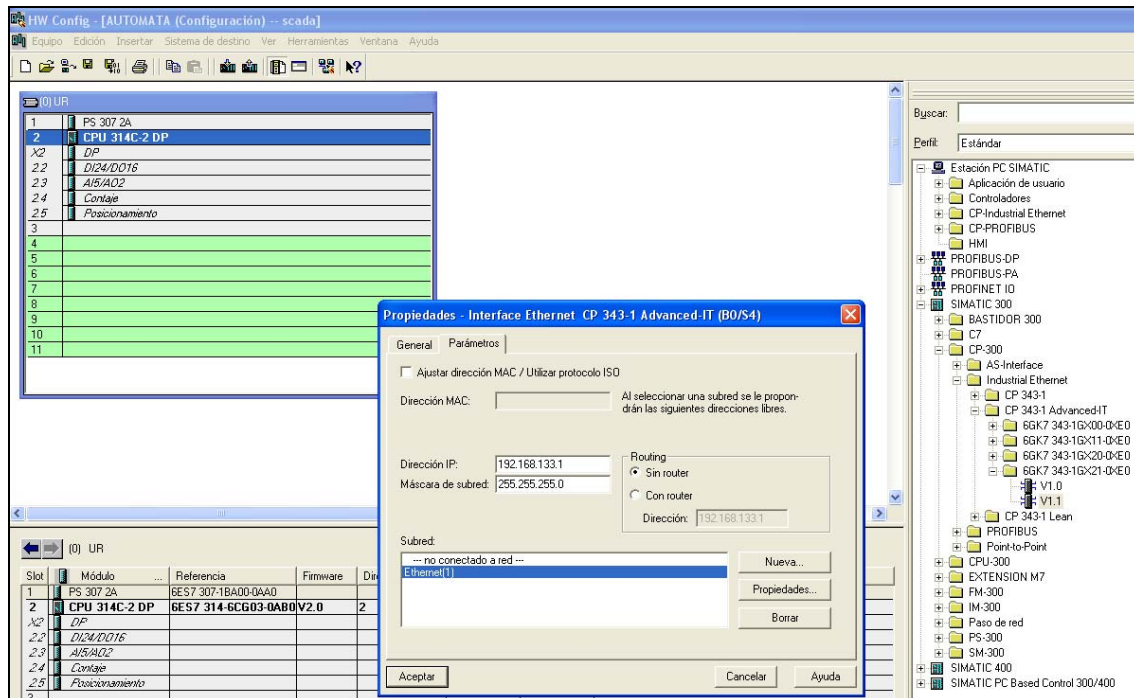


Figura 4.47. Configuración hardware del autómata, Procesador de Comunicaciones.

Al insertar el CP (Procesador de Comunicaciones) crearemos la red Ethernet. En la pantalla de propiedades que aparece en la imagen superior, pulsaremos en “Nueva” y aceptaremos, de forma que habremos creado la red Ethernet(1). En esta ventana también tendremos que fijar una dirección IP para el CP, que en nuestro caso es 192.168.133.1. Es importante tener en cuenta a partir de ahora que todos los elementos que se inserten en la red deberán tener la misma dirección IP variando únicamente el último dígito.

Para terminar con la configuración hardware del autómata, nos queda configurar el Switch Ethernet (módulo SCALANCE X208). Para ello, nos vamos a Sistema de destino→Editar estaciones Ethernet.

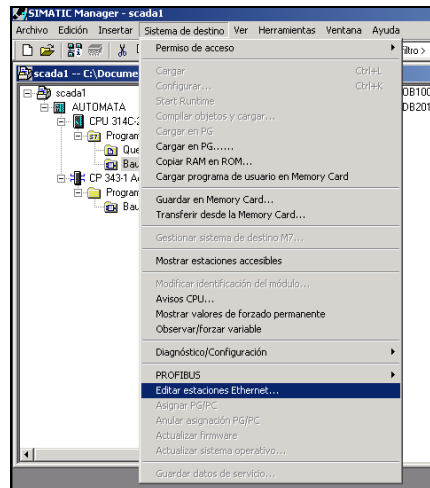


Figura 4.48. Editar estación Ethernet.

Desde aquí podremos detectar las estaciones online que tenemos accesibles en la red.

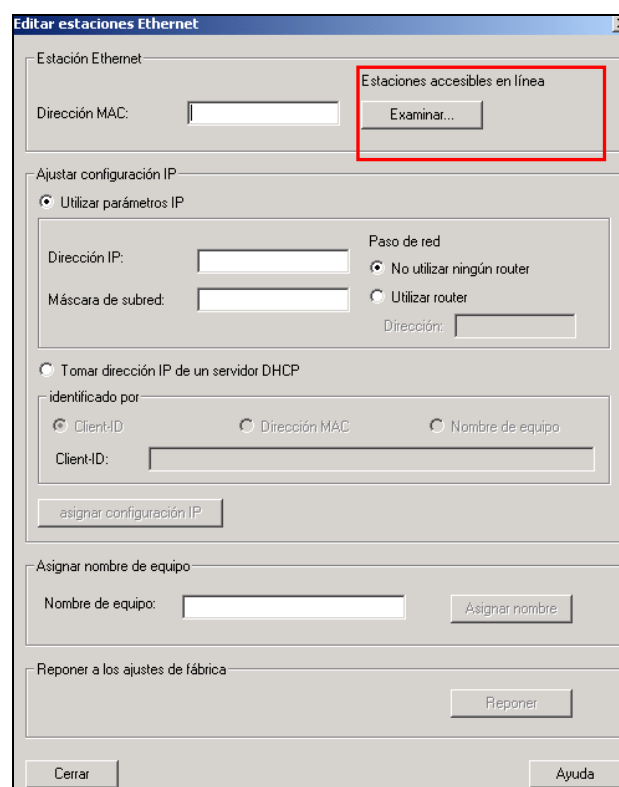


Figura 4.49. Buscar estaciones Ethernet online.

Aparecerán el procesador de comunicaciones y el switch Ethernet SCALANCE, al que le tendremos que asignar una dirección IP coherente, en este caso 192.168.133.5.

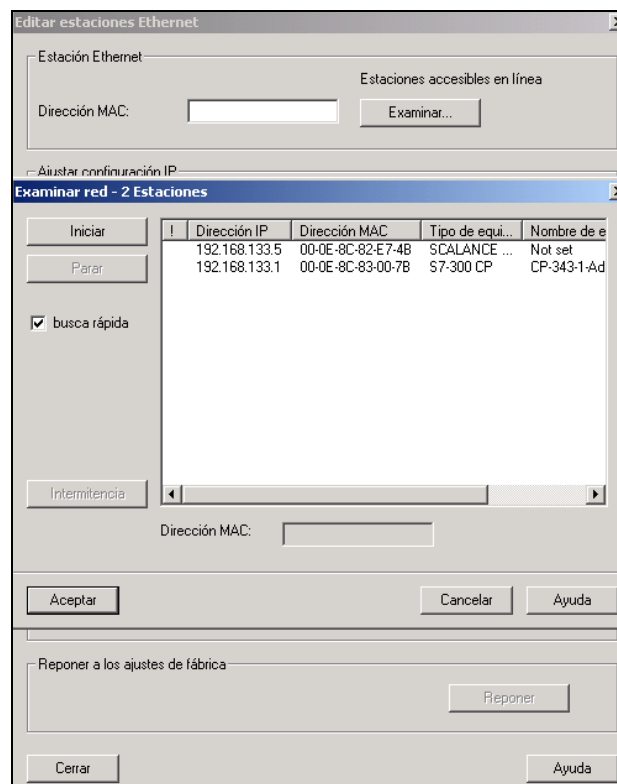


Figura 4.50. Estaciones Ethernet.

Por último, guardamos los cambios efectuados en la configuración del autómata.

Ahora pasamos a insertar el PC, que en este caso hemos llamado SCADA. Haremos clic sobre Insertar Nuevo Objeto → PG/PC.

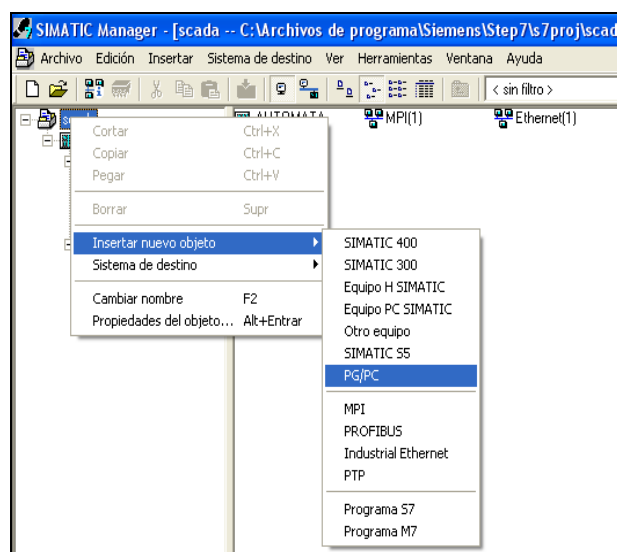


Figura 4.51. Insertar un PC en la red.

A continuación asignaremos la red Ethernet(1) creada anteriormente al PC. Haremos doble clic sobre el objeto PC y en la pestaña interfaces pulsaremos Nuevo→Industrial Ethernet.

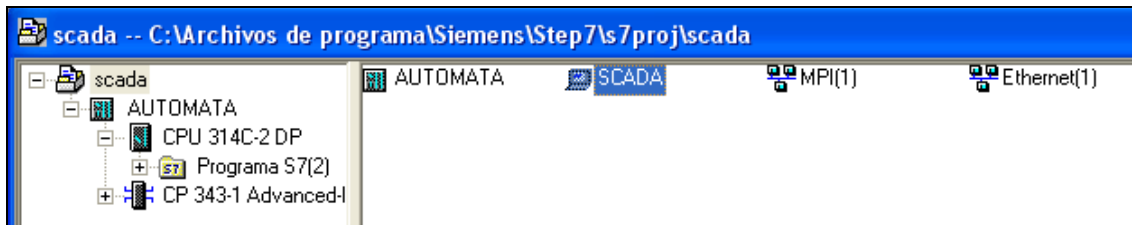


Figura 4.52. Conjunto de elementos insertados en la red.

Seleccionaremos la red Ethernet(1) e introduciremos la IP del PC, 192.168.133.3.

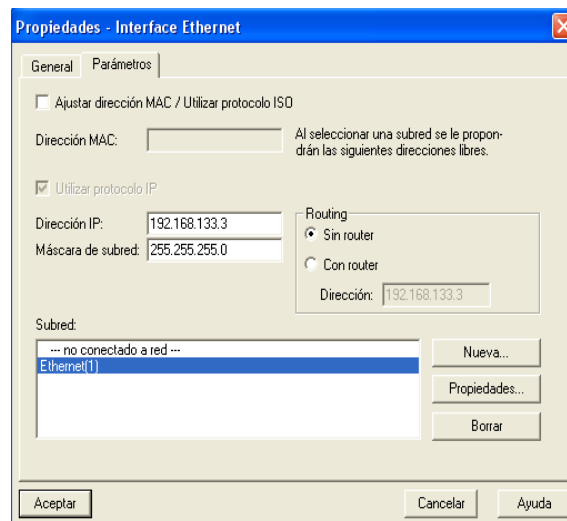


Figura 4.53. Asociar PC a red Ethernet.

Esta misma IP a su vez tendremos que asignarla en el PC de la siguiente forma:

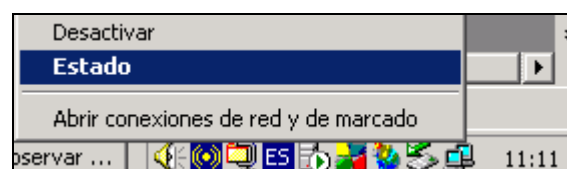


Figura 4.54. Conexión Ethernet PC.

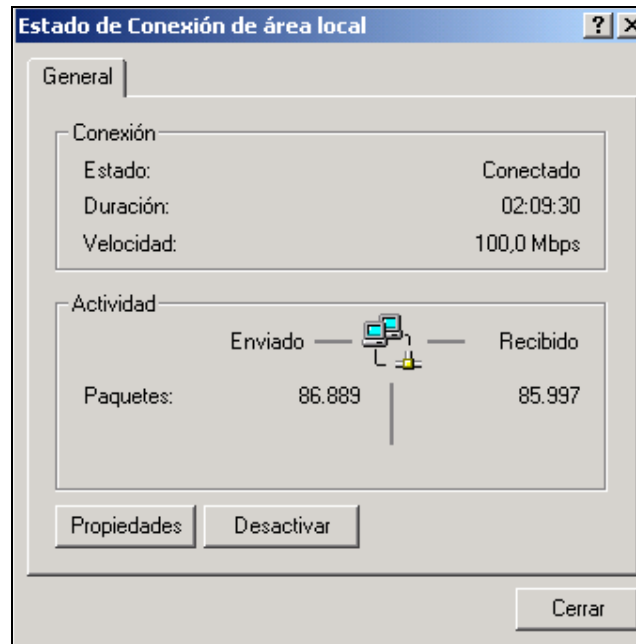


Figura 4.55. Estado conexión Ethernet PC.

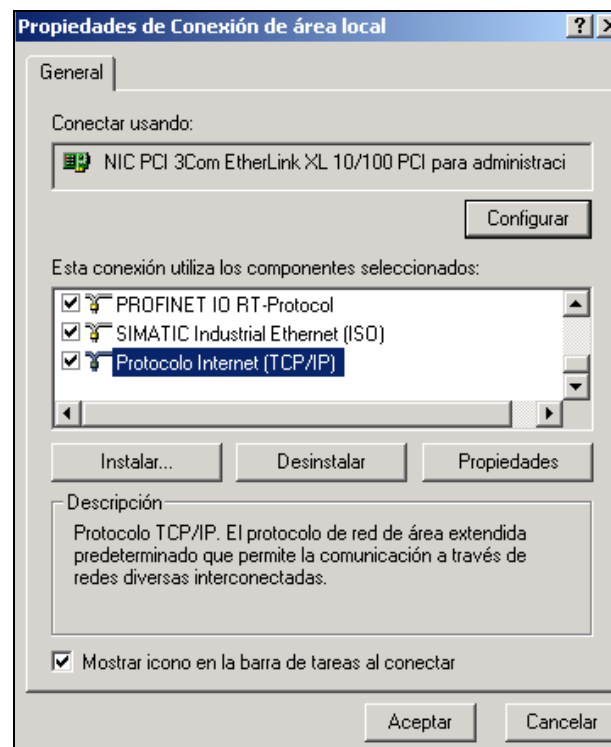


Figura 4.56. Protocolo TCP/IP Ethernet.

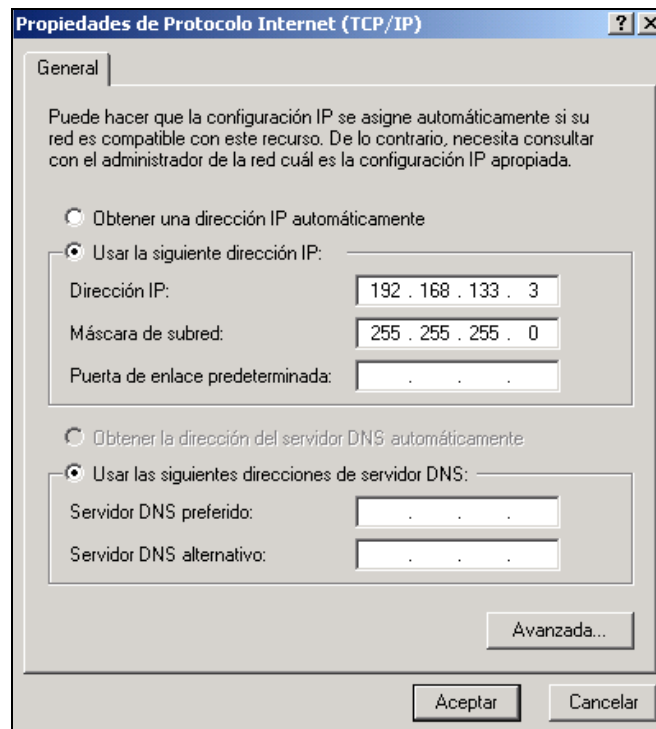


Figura 4.57. Propiedades del protocolo TCP/IP en el PC.

Una vez hecho esto, ya tenemos configurada la red Ethernet entre el PC y el autómata. Esto puede comprobarse en el Netpro, que se abre desde Herramientas→Configurar Red.

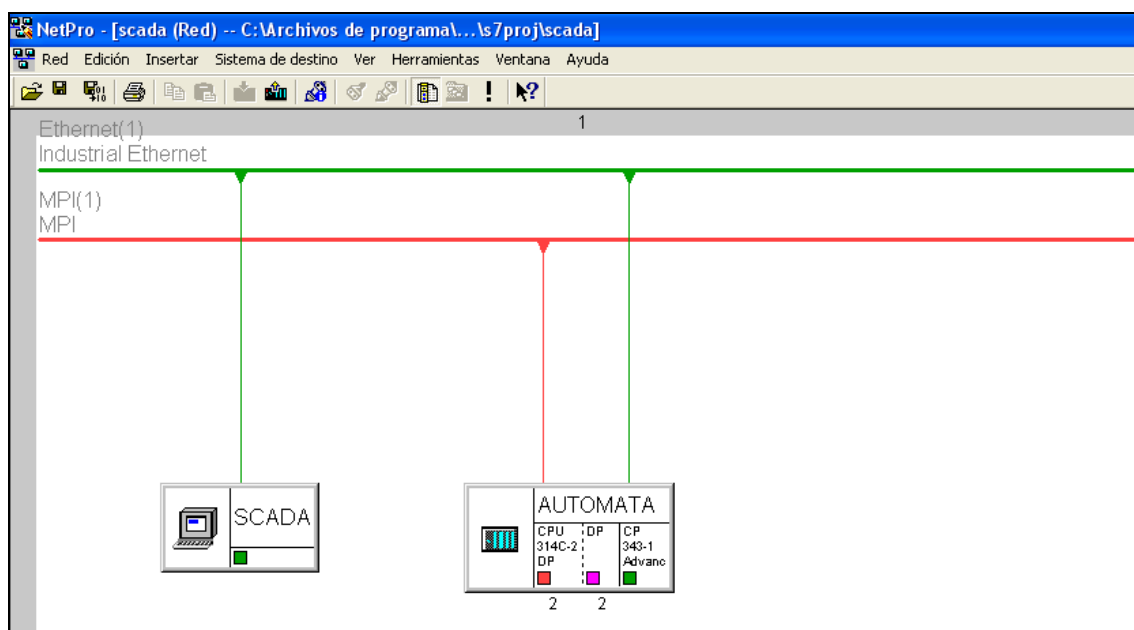


Figura 4.58. Red de comunicación entre el PC y el autómata.

4.2. Comunicación Autómata - Robot.

4.2.1. Objetivo, arquitectura y diagrama de flujo.

En este punto el objetivo es conseguir un intercambio de información básico entre el autómata Siemens S7 y el controlador de robot de ABB IRC5. Desde el autómata se enviará un byte almacenado en su memoria, y desde el IRC5 enviaremos una variable también de tipo byte previamente creada con el editor de variables del FlexPendant.

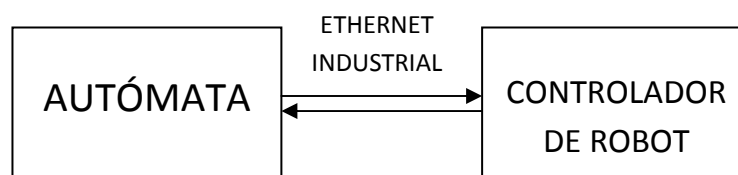


Figura 4.59. Comunicación Autómata – Robot.

La comunicación en el autómata se realiza mediante un programa implementado con el software de Siemens SIMATIC Manager. Dicho programa básicamente consiste en parametrizar adecuadamente los bloques de función que realizan las funciones de envío y recepción de datos. La comunicación en el controlador de robot se realiza a través de la interfaz *sockets*. Se trata de un canal de comunicación definido por direcciones IP, un protocolo (en nuestro caso TCP) y un número de puerto. Este proceso de comunicación brevemente descrito ahora, es explicado más detalladamente a lo largo de este apartado.

La arquitectura a implementar para lograr la comunicación Autómata - Controlador sería la siguiente:

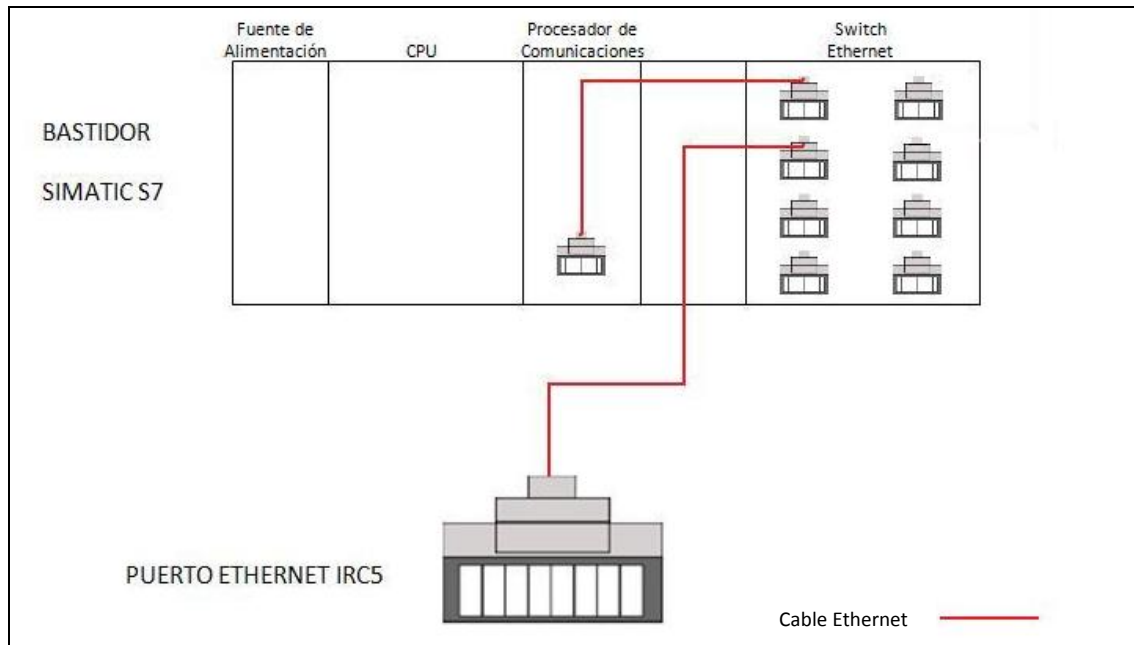


Figura 4.60. Arquitectura de comunicación Autómata – Robot.

En la siguiente página se muestra un diagrama de flujo para explicar el procedimiento de comunicación entre el autómata y el controlador de robot. Ambos programas, tanto el del autómata como el del controlador de robot, se ejecutarán de manera cíclica.

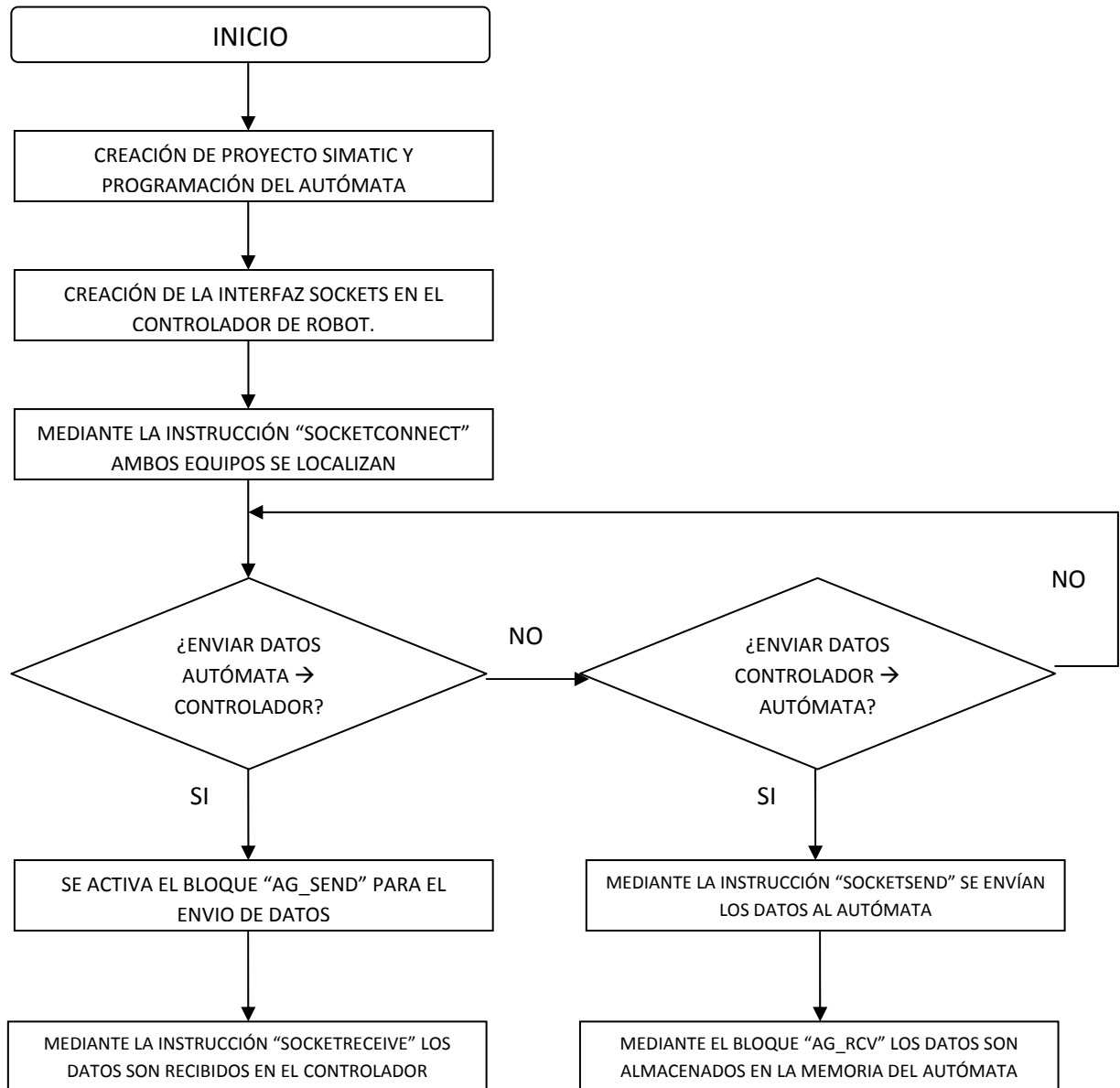


Figura 4.61. Diagrama de flujo de la comunicación Autómata – Robot.

Para establecer una comunicación entre el autómata y el robot, lo primero que se debe hacer es crear un proyecto en SIMATIC Manager para el autómata, y un programa en RAPID que use el canal de comunicación *sockets* en el controlador de robot. Una vez programadas las funciones de envío y recepción de datos como se explica en los apartados 4.2.3.1 y 4.2.3.2, el primer contacto entre ambos equipos tiene lugar al ejecutarse en el robot la instrucción "SocketConnect", con la cual los dos equipos son conectados empleando direccionamiento por IPs del mismo segmento.



Llegados a este punto, puede comenzar el envío y recepción de datos en ambos sentidos. Si se quieren enviar datos desde el autómata al controlador de robot, se utilizará el bloque AG_SEND en el PLC y la instrucción “SocketReceive” en el robot. En sentido contrario, el envío de datos en el robot se efectúa con “SocketSend” y su recepción en el autómata mediante el bloque de programa “AG_RCV”.

4.2.2. Conexión del robot a la red Ethernet. Creación de enlace TCP.

A la red de comunicación creada anteriormente entre el autómata y el PC tenemos que añadir un nuevo objeto: el controlador del robot IRC5. Para ello nos situamos sobre el nombre del proyecto y hacemos clic sobre Insertar nuevo objeto→Otro equipo, que renombraremos con el nombre IRC5.

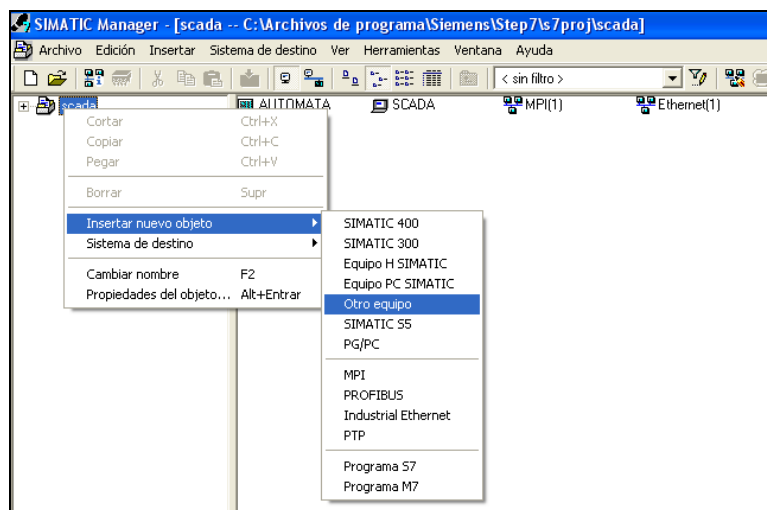


Figura 4.62. Insertar equipo ajeno (IRC5).

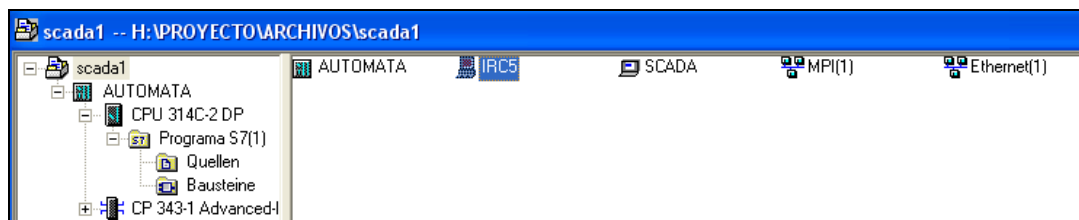


Figura 4.63. Vista de los elementos que componen la red.

A continuación lo que haremos es introducir el IRC5 en la red Ethernet(1). Para ello haremos doble clic sobre el objeto para acceder a sus propiedades, y en la pestaña “Interfaces”, pulsaremos en Nuevo→Industrial Ethernet.

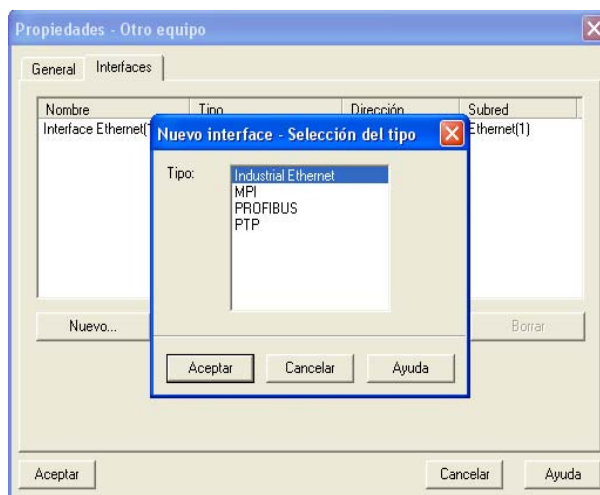


Figura 4.64. Creación de interfaz Industrial Ethernet para el IRC5.

Nos aparecerá una ventana en la que seleccionaremos la red Ethernet(1) y estableceremos la dirección IP del IRC5, que en este caso es 192.168.133.2, obtenida a través del FlexPendant.

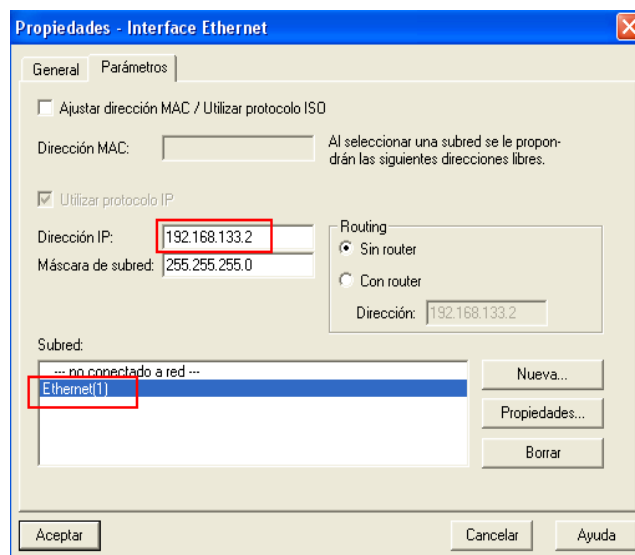


Figura 4.65. Asociación del IRC5 a la red Ethernet.



De esta forma, ya tendríamos integrados en la red Ethernet(1) los tres elementos que componen nuestra aplicación: PC, autómata y controlador de robot.

El último paso antes de dar por concluida la configuración de la red, es la creación de un enlace de comunicación en el autómata.

Concretamente se trata de un **enlace TCP no especificado**, lo cual significa que el PLC podrá comunicarse, siguiendo el protocolo TCP, con cualquier equipo en el que se trabaje también con el protocolo TCP, como es el caso del programa de sockets en lenguaje RAPID que se ejecutará en el robot y que es explicado en el punto 4.2.3.2.

El protocolo TCP garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto. TCP añade las funciones necesarias para prestar un servicio que permita que la comunicación entre dos sistemas se efectúe: libre de errores, sin pérdidas y con seguridad. Sus principales características son:

- ✓ Orientado a conexión: dos computadoras establecen una conexión para intercambiar datos. Los sistemas de los extremos se sincronizan con el otro para manejar el flujo de paquetes y adaptarse a la congestión de la red.
- ✓ Operación *Full-Duplex*: una conexión TCP es un par de circuitos virtuales, cada uno en una dirección. Sólo los dos sistemas finales sincronizados pueden usar la conexión.
- ✓ *Error Checking*: una técnica de *checksum* es usada para verificar que los paquetes no estén corrompidos.
- ✓ *Acknowledgements*: al recibir uno o más paquetes, el receptor devuelve un *acknowledgement* (reconocimiento) al transmisor indicando que recibió los paquetes. Si los paquetes no son notificados, el transmisor (emisor) puede reenviar los paquetes o terminar la conexión si se considera que el receptor no está conectado.

- ✓ *Flow Control*: si el transmisor está desbordando el buffer del receptor por transmitir demasiado rápido, el receptor descarta paquetes. Los *acknowledgement* fallidos alertan al receptor para bajar la tasa de transferencia o dejar de transmitir.
- ✓ Servicio de recuperación de Paquetes: el receptor puede pedir la retransmisión de un paquete. Si el paquete no es notificado como recibido (ACK), el transmisor envía de nuevo el paquete.

Para crear este enlace TCP, actuaremos de la siguiente forma:

Abriremos el Netpro en Herramientas→Configuración de red.

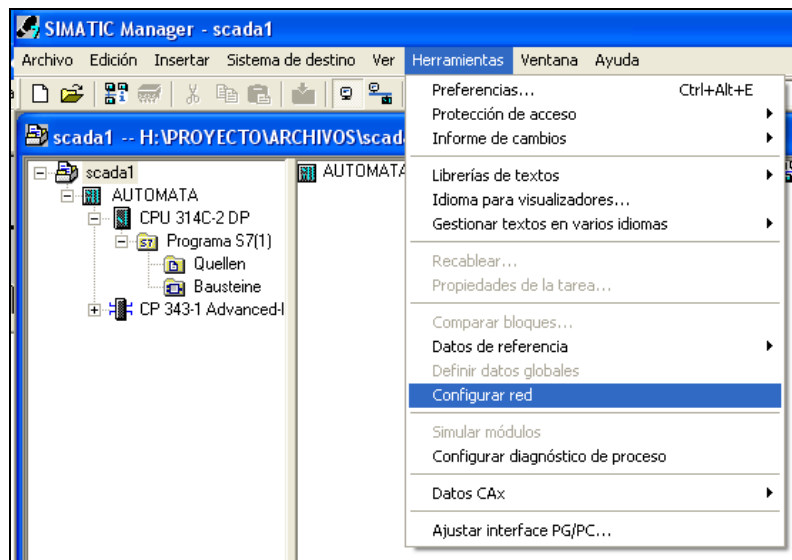


Figura 4.66. Abrir herramienta NetPro.

Luego nos situaremos sobre la CPU del autómata, pulsaremos el botón derecho del ratón y seleccionaremos “Insertar nuevo enlace”.

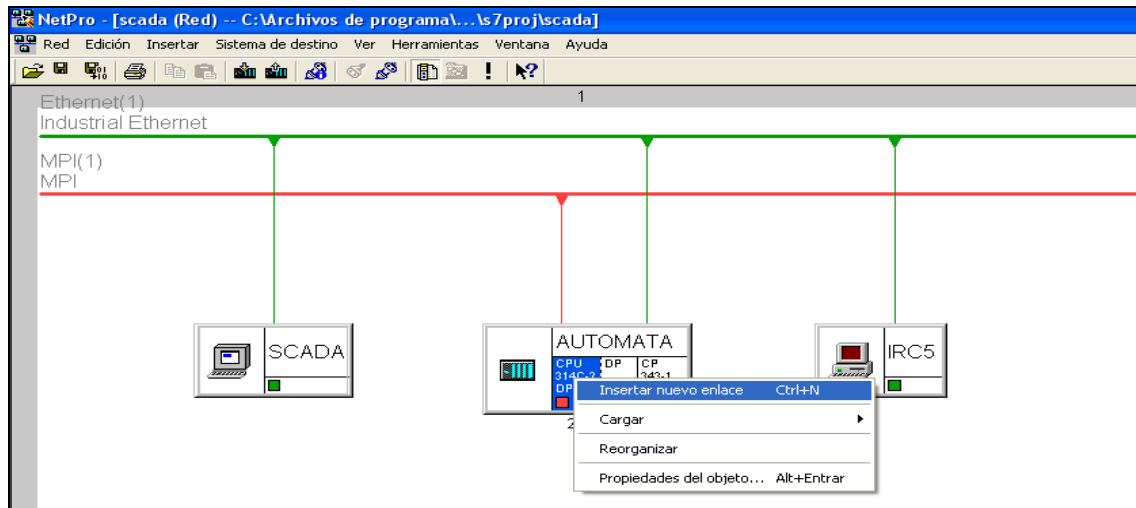


Figura 4.67. Creación de enlace TCP para comunicación con autómata.

Seleccionamos el tipo de enlace, TCP no especificado, y aceptamos.

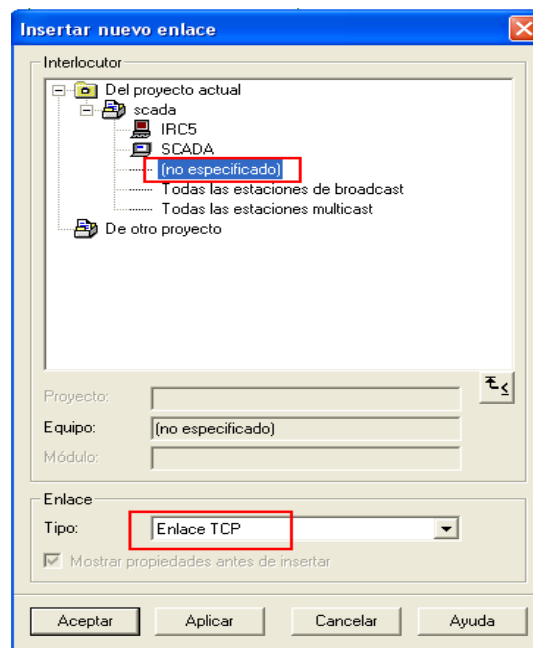


Figura 4.68. Selección del tipo de enlace.

Ahora nos aparecerá la ventana de propiedades del enlace.



En la pestaña “general” daremos un nombre al enlace y verificaremos que en Vía CP aparece el módulo CP Ethernet del autómata como interfaz de comunicación del PLC con el exterior.

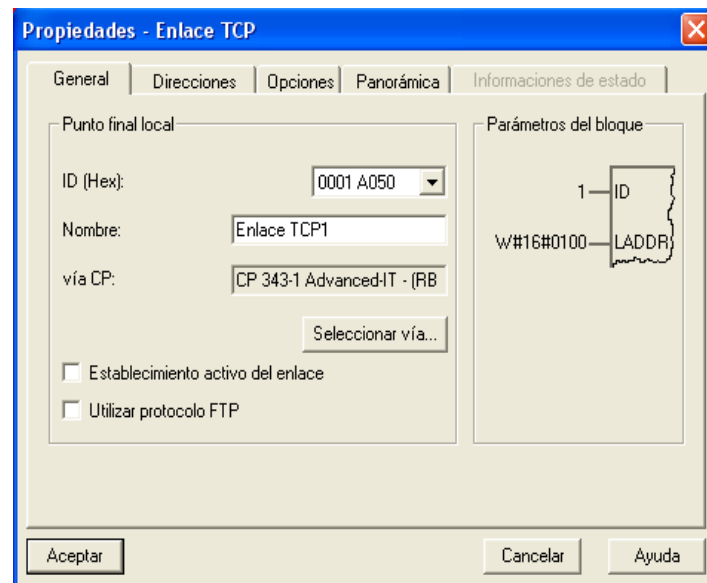


Figura 4.69. Propiedades del enlace TCP.

En la pestaña Direcciones, en la columna “Local”, deberá aparecer la IP del CP Ethernet. Aquí fijaremos con qué puerto el autómata se comunicará con los demás dispositivos mediante el protocolo TCP, en este caso hemos seleccionado el puerto 2000.

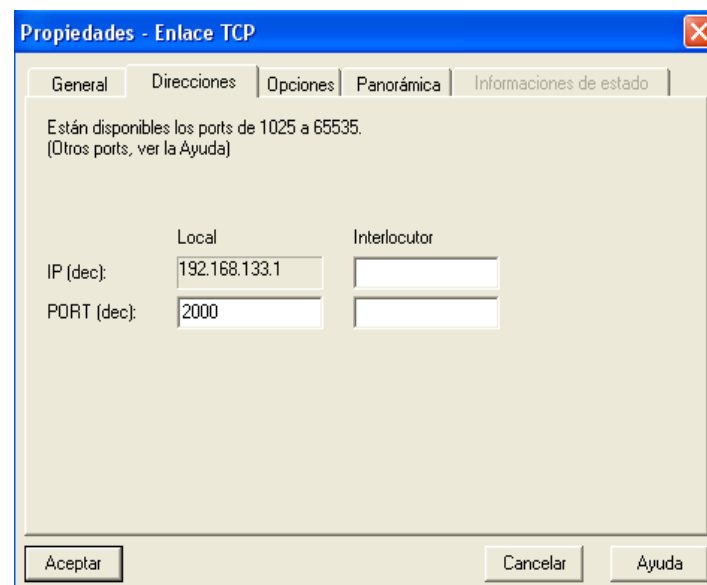


Figura 4.70. Direcciones de enlace TCP.



Dejaremos los campos del interlocutor sin completar, ya que se trata de un enlace no especificado, es decir, en teoría nos podemos comunicar con cualquier equipo de dirección IP 192.168.133.X que utilice el protocolo TCP.

Por último en la pestaña Opciones seleccionaremos el modo de operación local Send/Receive.

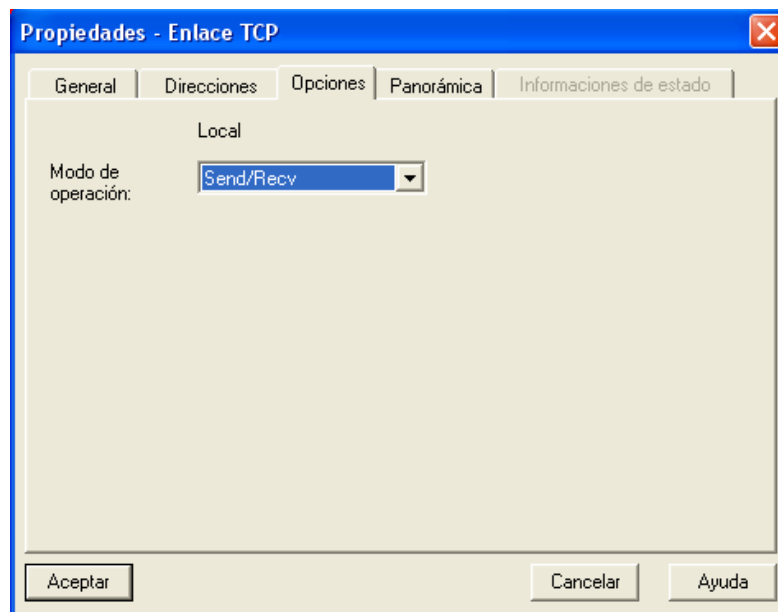


Figura 4.71. Selección de interfaz SEND/RECEIVE para el enlace TCP.

Podremos ver y editar el enlace creado en la tabla de la parte inferior del Netpro.

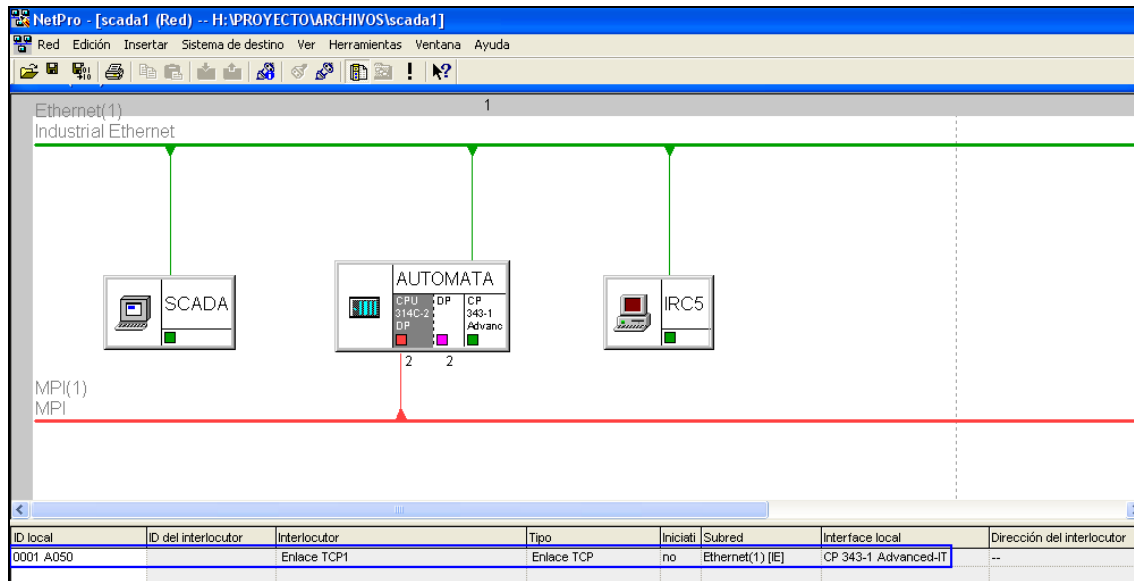


Figura 4.72. Herramienta de configuración de red Netpro.

Guardamos los cambios y compilamos, y pasamos a la programación del autómata en Step7.



Figura 4.73. Guardar y compilar la configuración del Netpro.

4.2.3. Envío y recepción de datos.

4.2.3.1. Autómata. Programación en Step7.

En la CPU del autómata deberá ejecutarse un programa que permita:

- El envío de datos, localizados en la memoria del autómata, hacia el robot.
- La recepción de los datos enviados por el robot.

Este programa ha sido desarrollado con el software de Siemens SIMATIC Manager. El lenguaje utilizado en esta memoria es el de contactos KOP. Si se prefiere, puede verse



el programa en los lenguajes AWL (lista de instrucciones) y FUP (diagrama de funciones) en el menú desplegable “Ver”:



Figura 4.74. Cambio del lenguaje de programación.

El programa del PLC está formado por una serie de bloques lógicos cuya estructura es la siguiente:

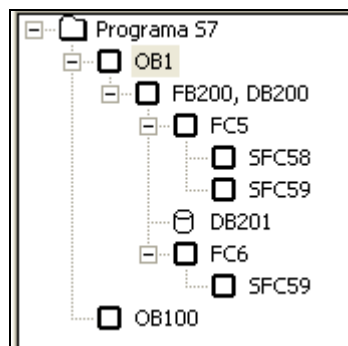


Figura 4.75. Estructura del programa Step7.

A continuación se expone brevemente en qué consiste cada bloque:

- **OB1:** Bloque de organización de programa. Se ejecuta cíclicamente en la CPU del autómata. En él se llama al bloque de función FB200, que a su vez lleva asociado el bloque de datos de instancia DB200, que contiene los parámetros con los que trabaja el FB200.



- **OB100:** Bloque de organización de programa que se ejecuta al reiniciar la CPU. En él se da permiso para que se inicie al proceso de comunicación tras una parada y puesta en marcha de la CPU.
- **FB200:** Bloque de función (bloques de libre programación por parte del usuario) en el cual se programan las funciones de envío (FC5 → AG_SEND) y recepción (FC6 → AG_RCV) de datos. Este bloque de función lleva asociado el bloque de datos **DB200**, el cual es instanciado únicamente por el FB200 siempre que éste es llamado. En el DB200 se encuentran todos los parámetros con los que se trabaja dentro del FB200 y la información relativa a los mismos (dirección, tipo de declaración, nombre simbólico, tipo de datos, valor inicial y valor actual).
- **FC5:** Función que permite el envío de datos desde el autómata. El bloque FC5 (AG_SEND) transfiere datos al CP Ethernet (Procesador de Comunicaciones) para su comunicación a través del enlace TCP configurado. Al finalizar su ejecución, indicará si el envío de datos se ha realizado correctamente o con errores. Los datos a enviar serán obtenidos del Bloque de Datos global DB201.
- **FC6:** Función que permite la recepción de datos en el autómata. Este bloque (AG_RCV) recibe del CP Ethernet los datos transferidos por el Controlador de Robot gracias al enlace TCP configurado. El FC6 informará cuando se reciban datos, y también nos dirá si éstos se han recibido correctamente o con errores. Los datos recibidos son almacenados en el Bloque de Datos global DB201.
- **DB201:** Bloque de Datos Global (se puede acceder a él desde cualquier bloque de función). Se trata de una estructura de datos de 200 bytes de longitud, en la que se encuentran el buffer de envío (byte 0) y el buffer de recepción (byte 100).
- **SFC58:** Función de Sistema integrada en la CPU del autómata utilizada para escribir registros ("WR_REC"). Se trata de una rutina llamada por el bloque FC5 "AG_SEND" cada vez que éste es ejecutado. Al ser una Función de Sistema sus parámetros no pueden ser editados. Este bloque es insertado



automáticamente por el software SIMATIC Manager tras insertar y parametrizar sin errores el bloque FC5.

- **SFC59:** Función de Sistema integrada en la CPU del autómata utilizada para escribir registros (“RD_REC”). Se trata de una rutina llamada por el bloque FC6 “AG_RCV” cada vez que éste es ejecutado. Al ser una Función de Sistema sus parámetros no pueden ser editados. Este bloque es insertado automáticamente por el software SIMATIC Manager tras insertar y parametrizar sin errores el bloque FC6.

Para obtener esta estructura de programa se seguirán estos pasos:

1. En la carpeta bloques, con el botón derecho, crearemos un bloque de organización OB1 y un bloque de función que denominaremos FB200, al cual automáticamente le será asociado el bloque de datos de instancia DB200 (accesible sólo por el FB200), que deberá rellenarse con la información de los parámetros con los que se trabaja en el FB200. También crearemos el bloque de datos global DB201 (accesible desde cualquier bloque).
2. Abrimos el bloque OB1, y en él insertaremos el FB200, que se encontrará en la carpeta “Bloques de Función” disponible en las librerías de bloques del SIMATIC Manager. Seguidamente le asociaremos el DB200.
3. Haciendo doble click accedemos al FB200, y en su interior insertaremos el bloque FC5 AG_SEND y el bloque FC6 AG_RECV disponibles en la librería SIMATIC_NET_CP → CP300. El buffer de envío y recepción se halla en el bloque de datos DB201 que, al igual que las funciones de sistema SFC58 y SFC59, será insertado automáticamente por el programa una vez que el FC5 y FC6 sean parametrizados correctamente tanto en sus campos de entrada como de salida.

Para actualizar la estructura de programa conforme se van añadiendo nuevos bloques puede usarse la opción “Actualizar Referencias Cruzadas”.

Si vamos a la carpeta “Bloques”, situada en el explorador del proyecto, podemos ver los archivos que lo componen. Desde aquí podremos abrir los distintos bloques para leerlos, crearlos o editarlos, siempre que no se trate de bloques protegidos por el sistema como son el FC5, el FC6, el SFC58 y EL SFC59.

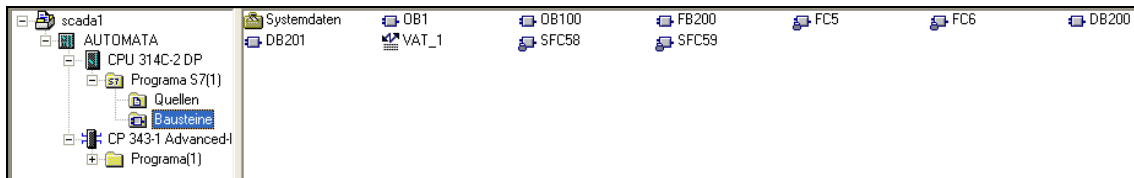


Figura 4.76. Bloques del programa implementado en el S7.

Puede apreciarse que se ha creado el objeto “VAT_1”. Se trata de una tabla de variables que sirve para forzar y visualizar el valor de los buffer de envío y recepción. Más adelante será tratada con más profundidad.

La parte **fundamental** del programa consiste en la llamada de los bloques de función **FC5 “AG_SEND”** y **FC6 “AG_RECV”**, cuyo empleo es posible sobre la base de la conexión TCP creada anteriormente. Podemos encontrar estos bloques en la librería **SIMATIC_NET_CP→CP 300**.

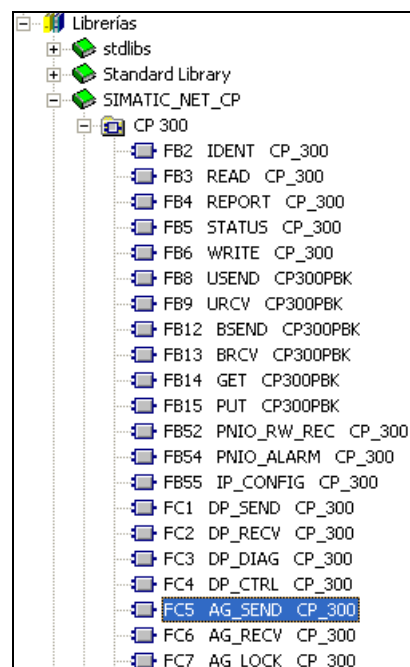


Figura 4.77. Bloques de la librería SIMATIC NET CP.



El bloque AG_SEND sirve para enviar datos a otro autómata S7, a un PC o a un tercer dispositivo (como es el IRC5). Asimismo, el bloque AG_RECV sirve para recibir datos de otro autómata S7, de un PC o de un tercer dispositivo.

Lo primero que tenemos que hacer es establecer el byte de marcas 10 como marca de ciclo, ya que el Timer T10 es utilizado por los bloques del programa. Para ello abriremos la configuración hardware del autómata.



Figura 4.78. Abrir configuración hardware del autómata.

Abrimos las propiedades de la CPU con el botón derecho, y en la pestaña “Ciclo/Marca de ciclo” fijamos el byte de marcas 10 como marca de ciclo. Una vez hecho esto, guardaremos, compilaremos y cargaremos la configuración hardware en el autómata.

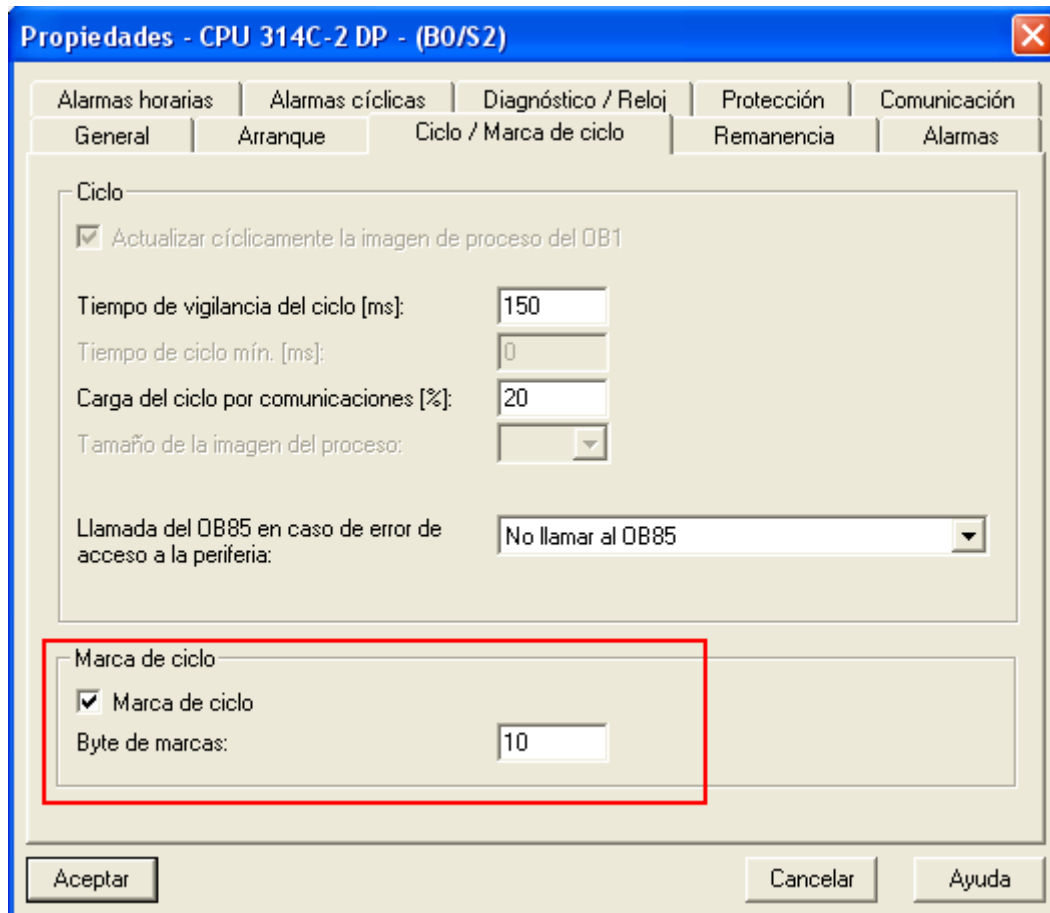


Figura 4.79. Configuración del byte 10 como marca de ciclo.

Una vez hecho esto, guardamos los cambios realizados.

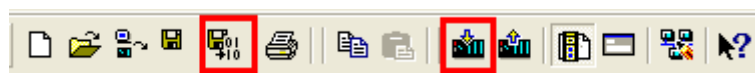


Figura 4.80. Guardar, compilar y cargar la configuración hardware en el PLC.

A la hora de parametrizar los bloques AG_SEND y AG_RECV, los campos "ID" y "LADDR" deben ser obtenidos del enlace TCP creado, al cual accedemos desde el Netpro como se ha explicado con anterioridad.

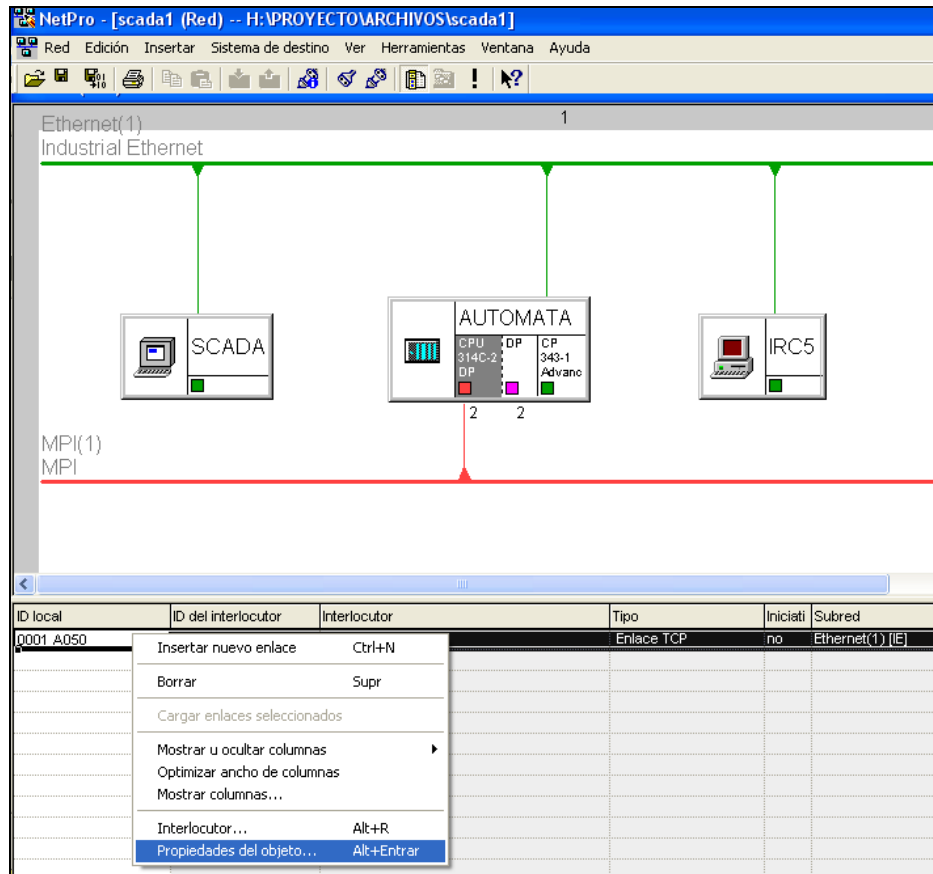


Figura 4.81. Propiedades del enlace TCP en el Netpro.

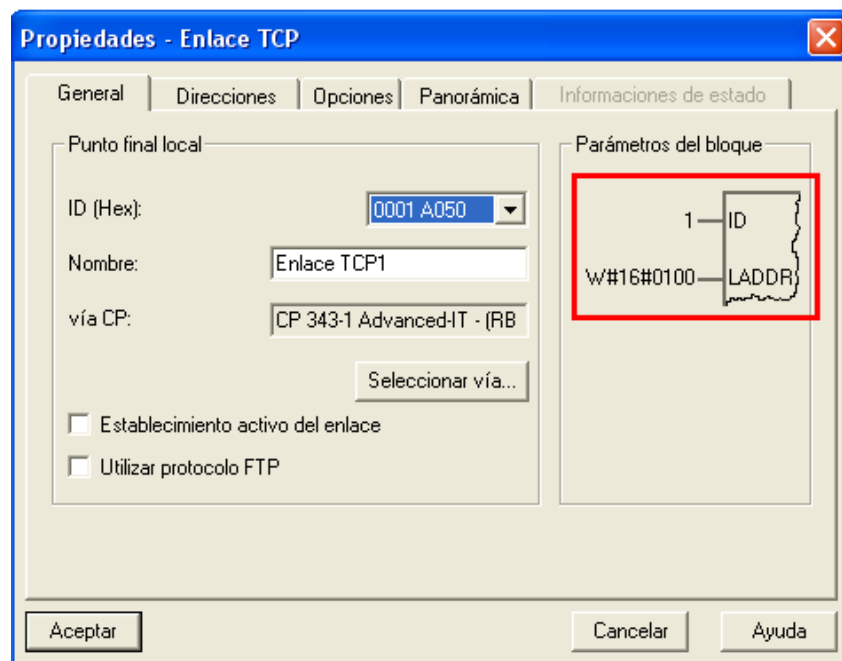


Figura 4.82. Obtención de parámetros ID y LADDR del enlace TCP.



Llegados a este punto, pasamos a describir los bloques del programa implementado.

OB100:

Se trata de un bloque que se ejecuta cuando la CPU es reiniciada. El bit M0.3 "START-UP" será usado para indicar el inicio de la comunicación en el bloque OB1.

OB100 : Bloque de Organización OB100

Este bloque se ejecuta al reiniciar la CPU del autómata.

Segm. 1: Título:

Cualquiera que sea el valor de M0.1 --> Activamos el bit "START-UP" de inicio de la comunicación



OB1:

Este bloque se ejecuta cíclicamente. Cuando el bit START-UP=1 se produce la llamada al Bloque de Función FB200 "AG_SEND/AG_RECV", que lleva asociado el bloque de datos de instancia DB200, donde se encuentran los nombres simbólicos de los parámetros usados en dicho bloque.

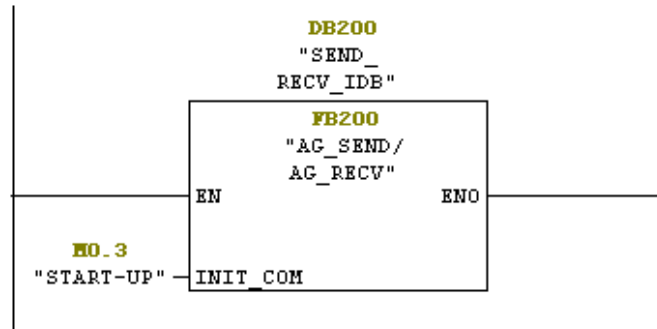


OB1 : Bloque de ejecución cíclica

Comentario:

Segm. 1: Título:

Llamamos al Bloque de Función FB200, que lleva asociado el Bloque de Datos DB200 donde se encuentran los parámetros con los que se trabaja en el FB200. Dentro de este bloque se realizarán las funciones de envío y recepción de datos.



Segm. 2: Título:

Sólo queremos llamar una vez al bloque FB200



FB200:

En este bloque se produce la llamada de los bloques FC5 “AG_SEND” y FC6 “AG_RECV”.

La petición de envío es disparada con un flanco positivo del bit del parámetro ACT, en este caso el A124.1. Esta petición de envío es controlada por la marca de reloj M10.7 y la variable “SND_BUSY”. Si el envío está en curso, SND_BUSY está activado e intentar realizar un nuevo envío no es posible. Este bloqueo es importante para evitar posibles sobrecargas en la red de comunicación.

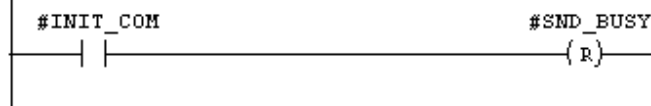


FB200 : Título:

Bloque para el ENVÍO y RECEPCIÓN de datos

Segm. 1 : Título:

Al iniciarse la comunicación --> Reseteamos el bit que indica si el envío está en curso o no.



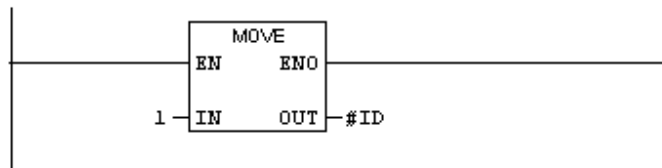
Segm. 2 : Título:

Con el bit STATISTIC ON/OFF elegimos si queremos monitorizar o no los parámetros de salida STATUS de los bloques AG_SEND y AG_RECV



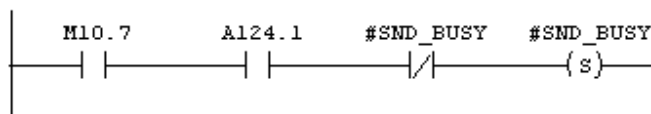
Segm. 3 : Título:

Escribimos el valor "1" en el campo ID necesario para la ejecución correcta de los bloques AG_SEND y AG_RECV



Segm. 4 : Start AG_SEND function with rising edge (clock marker MB10)

Si la marca de reloj M10.7 experimenta un flanco de subida, activamos el envío de datos (A124.1=1) y el bloque AG_SEND no está trabajando --> Activamos el bit SND_BUSY para indicar que estamos realizando un envío de datos.



Como se ha comentado anteriormente, el valor de los parámetros ID y LADDR debe ser obtenido del diálogo de propiedades del enlace TCP en el Netpro.

En el parámetro de entrada SEND del bloque AG_SEND se indica la dirección y longitud del área de datos a enviar (buffer de envío), mediante un puntero. En el parámetro de entrada LEN indicamos el número de bytes que queremos enviar.



En nuestro caso:

SEND = P#DB201.DBX0.0 BYTE 100

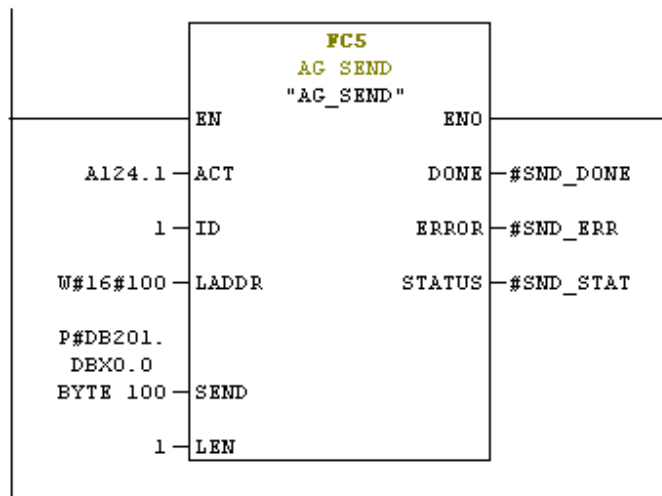
LEN = 1

Esto significa que enviaremos el primer byte de los 100 que componen el bloque de datos DB201.

Los parámetros de salida DONE, ERROR y STATUS se utilizan para evaluar si el proceso de envío se ha realizado correctamente o ha surgido algún error. Sólo son válidos durante un ciclo.

Segm. 5 : Título:

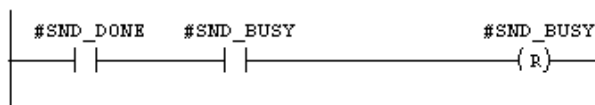
Llamada a la función de envío de datos "AG_SEND"





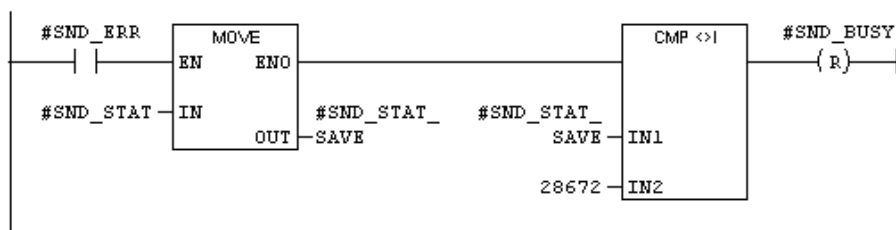
Segm. 6 : Título:

Si el envío se ha completado, reseteamos SND_BUSY



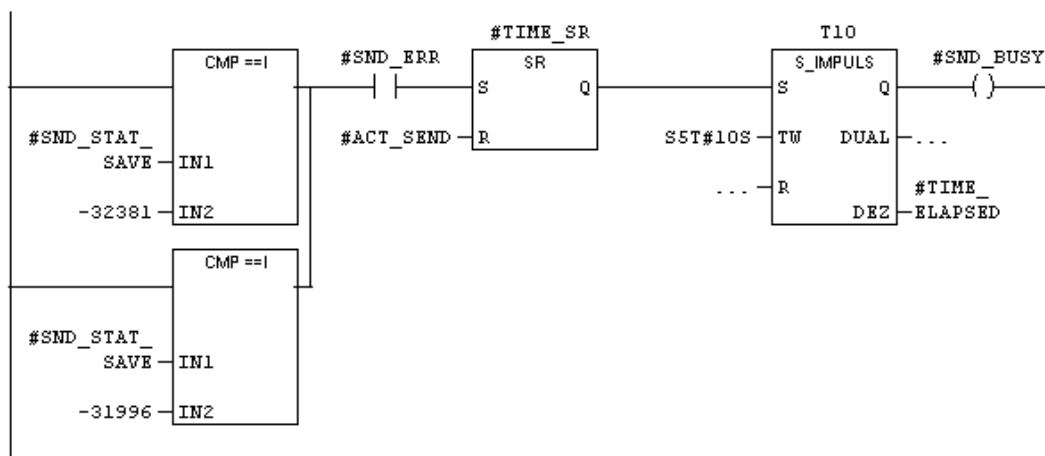
Segm. 7 : Evaluación del parámetro STATUS

Si se produce ERROR, guardamos STATUS y reseteamos BUSY



Segm. 8 : Si hay error de conexión, esperar 10s antes de invocar AG_SEND

STATUS = 0x8183 = -32381 decimal -> Conexión no establecida o ha sido desconectada.
= 0x8304 = -31996 decimal -> Conexión no establecía, sin actualizar.



A continuación invocamos el bloque AG_RECV. Los parámetros ID y LADDR, al igual que para el bloque AG_SEND, serán adoptados de las propiedades del enlace TCP configurado en el Netpro. En el parámetro de entrada RECV especificaremos la dirección y longitud del buffer de recepción, mediante un puntero:

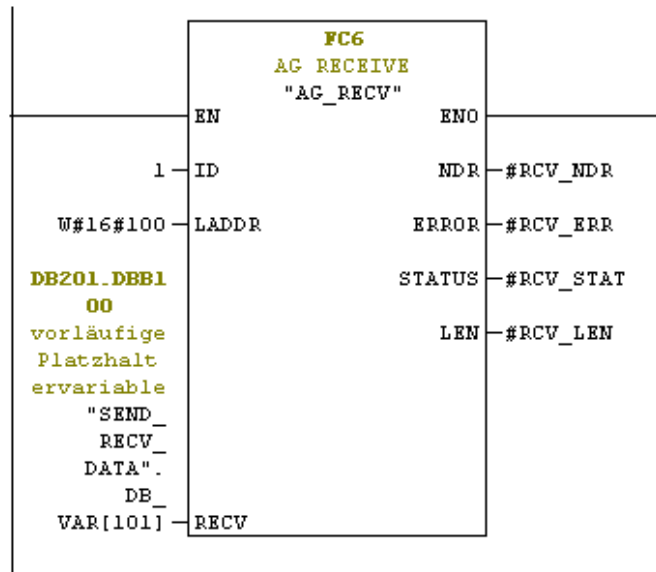
P#DB201.DBX100.0 BYTE 1



Según esto, puede verse que nuestra área de recepción será el byte DB201.DBB100. Se ha establecido una longitud de 1 byte porque lo que enviamos desde el controlador del robot es 1 byte.

Segm. 9: Llamada al bloque AG_RECV

Comentario:

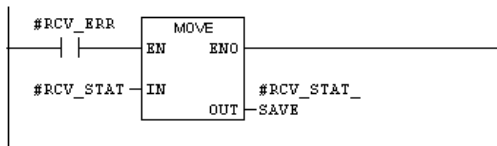


El parámetro de salida NDR sirve para mostrar que se han recibido nuevos datos. LEN indica la longitud de los datos recibidos.

Si los datos no son recibidos correctamente, guardamos STATUS y lo evaluamos.

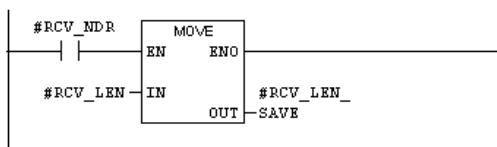
Segm. 10: Título:

Si la función nos da ERROR, guardamos STATUS



Segm. 11: Nuevos datos recibidos

Si recibimos nuevos datos, guardar la longitud de éstos (LEN)



Para poder leer y escribir en las posiciones de memoria del autómata donde se reciben y envían los datos, se ha creado una tabla de variables VAT_1, situada en la carpeta bloques de la CPU.

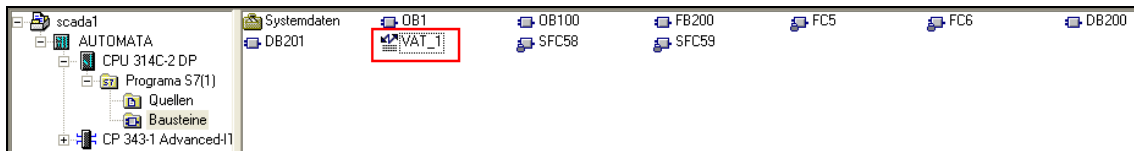


Figura 4.83. Tabla de variables creada.

Si accedemos a dicha tabla, podemos observar dos posiciones:

- DB201.DBB 0, byte de envío.
- DB201.DBB 100, byte de recepción.

| VAT_1 -- scada1\AUTOMATA\CPU 314C-2 DP\Programa S7(1) | | | | | |
|---|---------------|------------------------------|----------------|-----------------|------------------|
| | Operando | Símbolo | Formato de vis | Valor de estado | Valor de forzado |
| 1 | DB201.DBB 0 | "SEND_RECV_DATA".DB_VAR[1] | DEC | | 100 |
| 2 | DB201.DBB 100 | "SEND_RECV_DATA".DB_VAR[101] | DEC | | |

Figura 4.84. Buffer de envío y recepción; longitud 1 byte.

Como puede verse, desde esta ventana escribiremos (valor de forzado) y leeremos (valor de estado) los datos intercambiados con el controlador de robot.

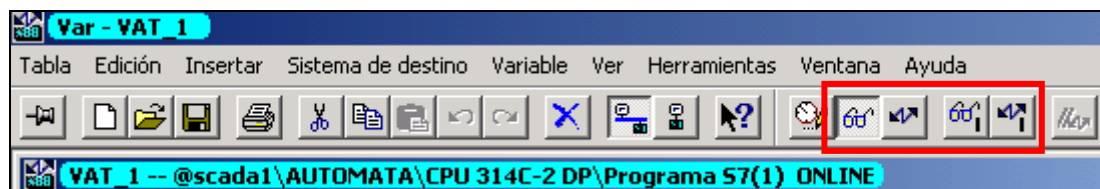


Figura 4.85. Valores de estado y forzado de los datos comunicados.

En el ejemplo de la figura 4.84, vemos que el valor de forzado del byte de envío es 100, por lo que en el FlexPendant veremos lo siguiente:

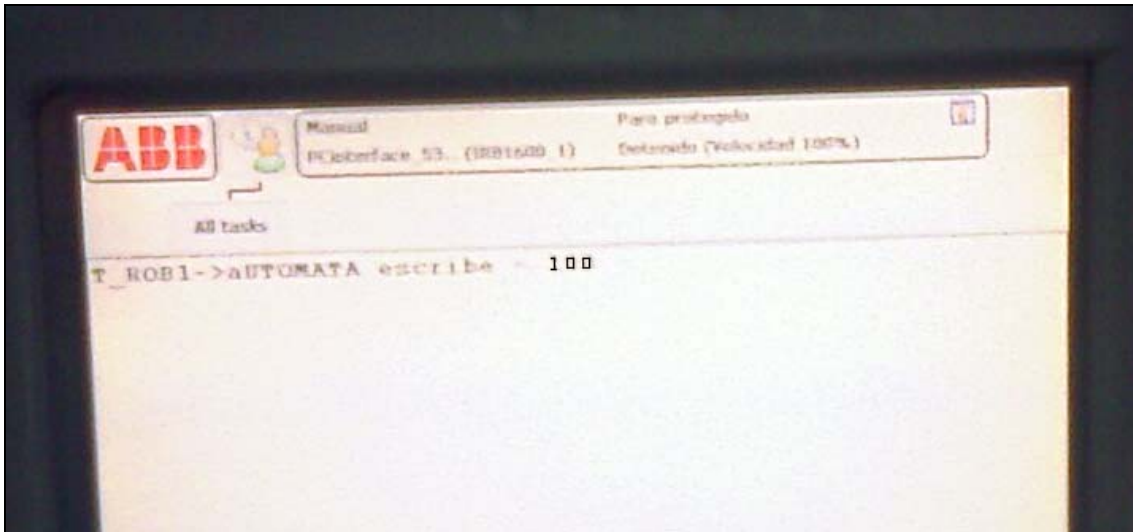


Figura 4.86. Recepción de datos en el FlexPendant del IRC5.

Llegados a este punto ya hemos terminado de crear la red con el software SIMATIC Manager. Para asegurarnos de que todas las configuraciones y programas se cargan en el autómata, procederemos de la siguiente forma:

1. Sobre el nombre del proyecto, botón derecho → Sistema de destino → Cargar y compilar objetos.
2. Sobre la carpeta “Bloques” → Sistema de destino → Cargar y compilar objetos.

Para entender cómo se envían y reciben datos en el controlador del robot, vamos a estudiar a continuación la programación en RAPID del IRC5.

4.2.3.2. Robot. Programación en RAPID mediante Sockets.

En el controlador de robot se ejecutará un programa en lenguaje RAPID que nos permitirá enviar datos al autómata y recibir datos de éste. Esto lo haremos usando el protocolo TCP mediante la opción que nos ofrece el software del IRC5 llamada “Socket Messaging”.

Como se ha comentado, la finalidad de la opción Socket Messaging es permitir a un programa de RAPID intercambiar mensajes de TCP/IP a través de una red. Esta utilidad



envía y recibe mensajes a través del canal Ethernet permanente del IRC5 (que puede usarse simultáneamente para otro tráfico de la red, por ejemplo para la comunicación con el software RobotStudio Online).

Los mensajes por socket constituyen un estándar admitido por sistemas operativos como UNIX o Microsoft Windows. Una comunicación de este tipo podemos dividirla en tres fases:

- Creación y cierre del canal de comunicación.
- Establecimiento de la sesión de comunicación.
- Envío y recepción de datos.

El siguiente módulo de RAPID contiene el programa implementado en nuestro caso. En él pueden distinguirse las tres operaciones anteriores. El diagrama de flujo que describe su funcionamiento es el siguiente:

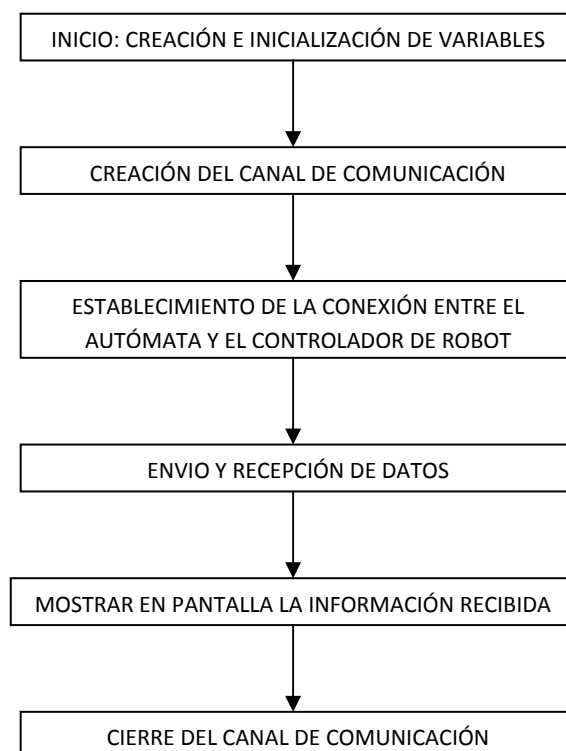


Figura 4.87. Diagrama de flujo del programa RAPID.



MODULE MainModule

```
VAR socketdev socket1; ! Variable utilizada para crear el socket

VAR string string1:=""; ! Var. para mostrar por pantalla el dato recibido

VAR byte env{1}:=[0]; ! Variable que enviamos. El SCADA leerá su valor

! y en función de éste se mostrará una imagen de la tarea que el robot lleva a cabo

VAR byte rec{1}:=[0]; !Variable donde guardamos el byte recibido

VAR num i:=0;

PROC main()

    env{1}:=0;

    SocketCreate socket1; ! Creamos el canal de comunicación

    SocketConnect socket1, "192.168.133.1", 2000; ! Conect. PLC pto. 2000

    WaitTime 5; ! Para evitar posibles problemas de comunicación

    FOR i FROM 0 TO 48 STEP 1 DO ! Tenemos 49 estados del robot

        SocketSend socket1\Data:=env;

        WaitTime 1;

        Add env{1},1;

        WaitTime 1;!Para poder ver los cambios en la animación SCADA

    ENDFOR

    SocketReceive socket1\Data:=rec; ! Recibimos los datos

    string1 := ByteToStr(rec{1}); ! Conversión de formato de datos

    TPWrite "AUTÓMATA escribe - " + string1; ! Mostrar en pantalla

    WaitTime 5;

    SocketClose socket1; ! Cerramos el canal de comunicación

ENDPROC

ENDMODULE
```



Se trata de un código para una aplicación de cliente que conectará con el servidor, es decir, con el autómata (cuya dirección IP es 192.168.133.1). Es importante recordar que el enlace configurado en el autómata no especificaba la dirección IP de su interlocutor. Esto significa que el PLC permanece “a la escucha” de los mensajes que le lleguen desde cualquier dispositivo.

4.3. Integración en el sistema PC-Autómata-Robot.

En los puntos anteriores se ha estudiado por separado primero la comunicación PC-Autómata y posteriormente la comunicación Autómata – Controlador de robot. Ahora veremos cómo interactúan, a nivel de sistema, estos tres componentes.

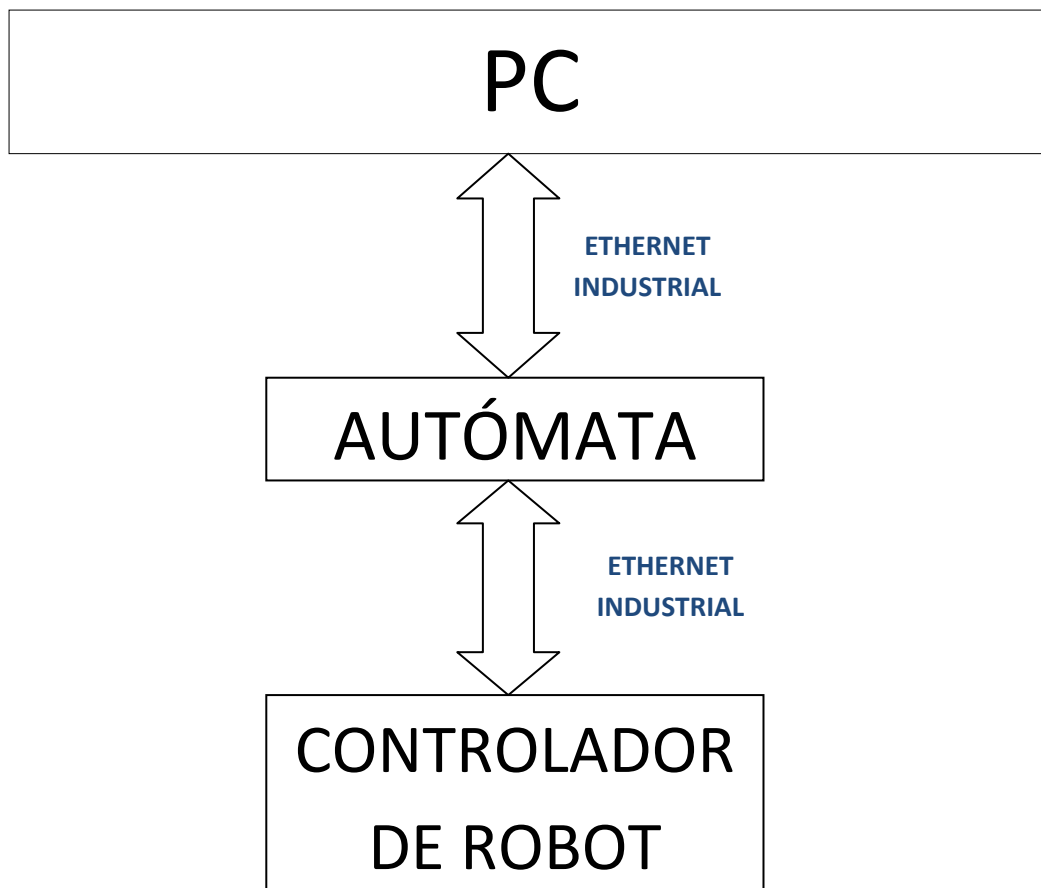


Figura 4.88. Esquema de bloques e interfaces de comunicación del sistema.

4.3.1. Esquema de conexiones del sistema.

A continuación se muestra un esquema del conexionado de todo el sistema, incluyendo los tres componentes, mediante la interfaz Ethernet Industrial.

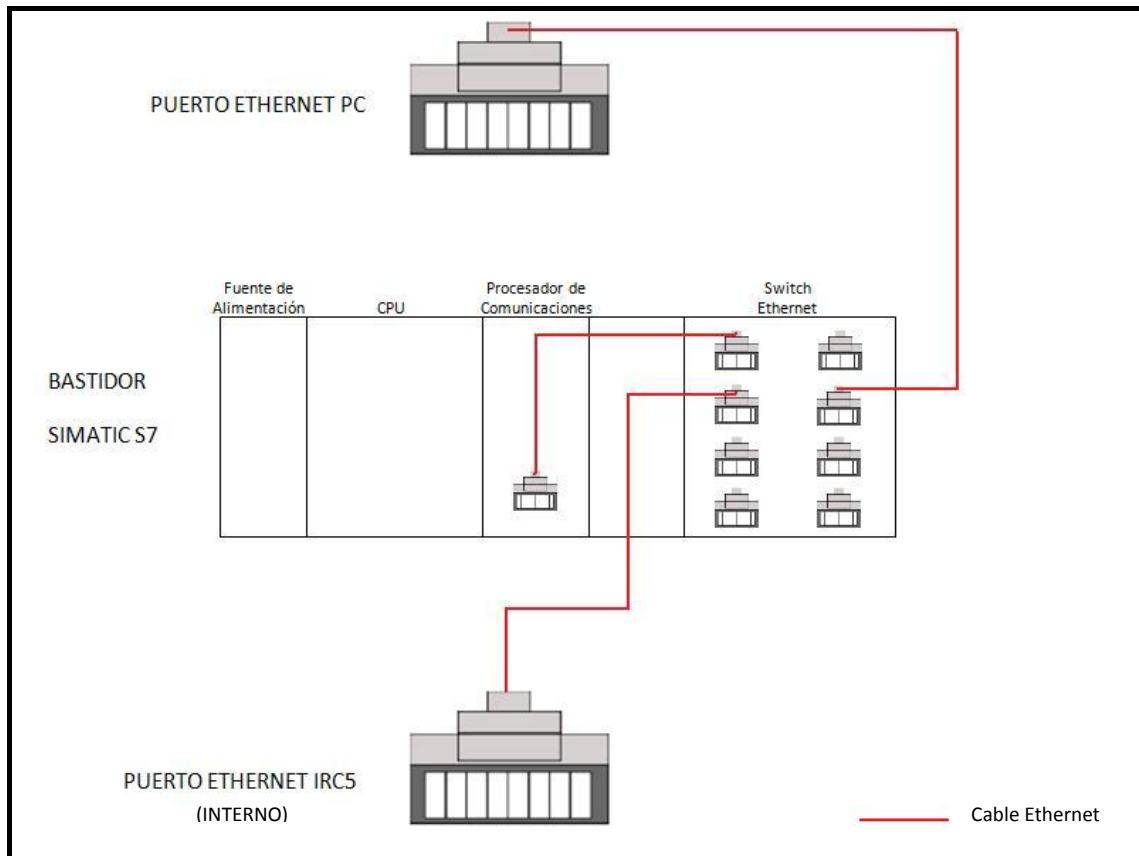


Figura 4.89. Esquema de conexiones; interfaz Ethernet Industrial.

El envío de datos entre dispositivos es bidireccional, es decir, tanto en el PC como en el IRC5 podemos enviar y recibir datos. Se trata de una red que funciona en modo *full dúplex* ya que permite canales de envío y recepción simultáneos.

4.3.2. Diagramas de flujo.

Para explicar el envío de datos desde **el PC hacia el controlador** planteamos el siguiente diagrama de flujo:

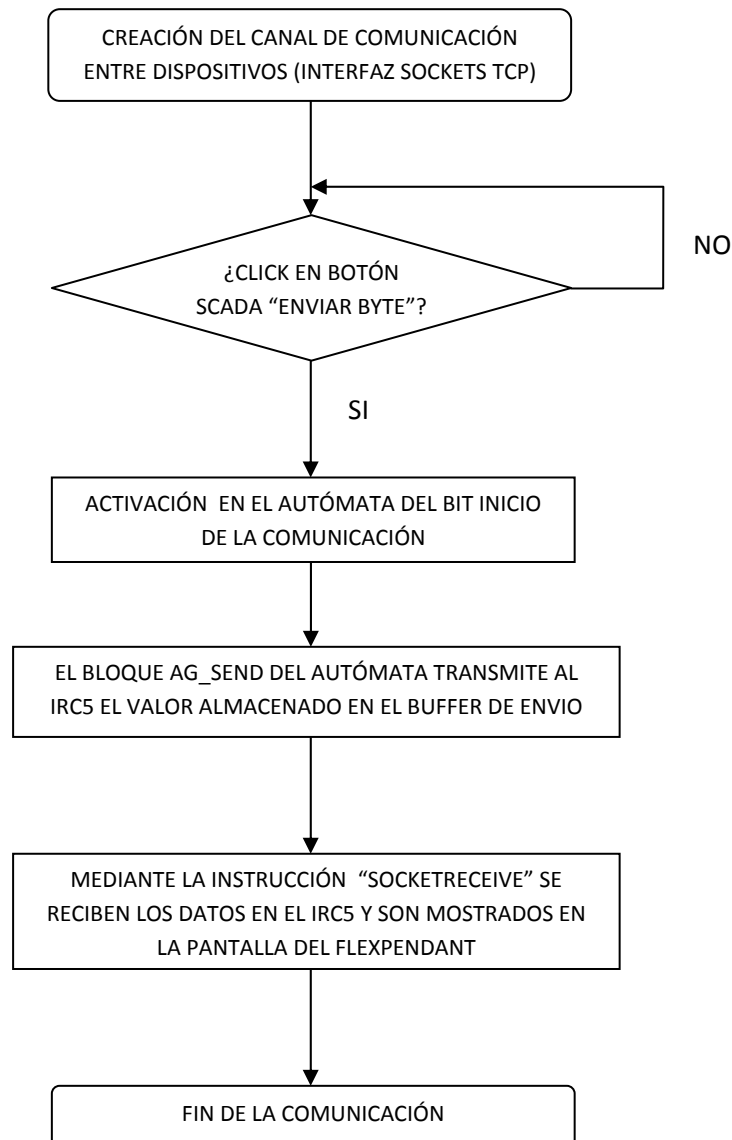


Figura 4.90. Diagrama de flujo; Envío de datos PC → Controlador de robot.

Una vez que se ha creado la red Ethernet Industrial entre los tres dispositivos (que se encontrarán conectados mediante la interfaz *sockets* y un enlace TCP), la comunicación en el PC se inicia al hacer click sobre el botón del SCADA “ENVIAR BYTE” (ver figura 4.93). Al pulsar sobre este botón se activa un bit que provocará que el bloque “AG_SEND” del autómata transmita al controlador de robot el dato almacenado en el buffer de envío. Estos datos serán recibidos en el robot a través de la instrucción “SocketReceive” y mostrados en pantalla mediante la instrucción “TPWrite”.

En sentido contrario, para enviar datos **desde el controlador hacia el PC** el proceso es el siguiente:

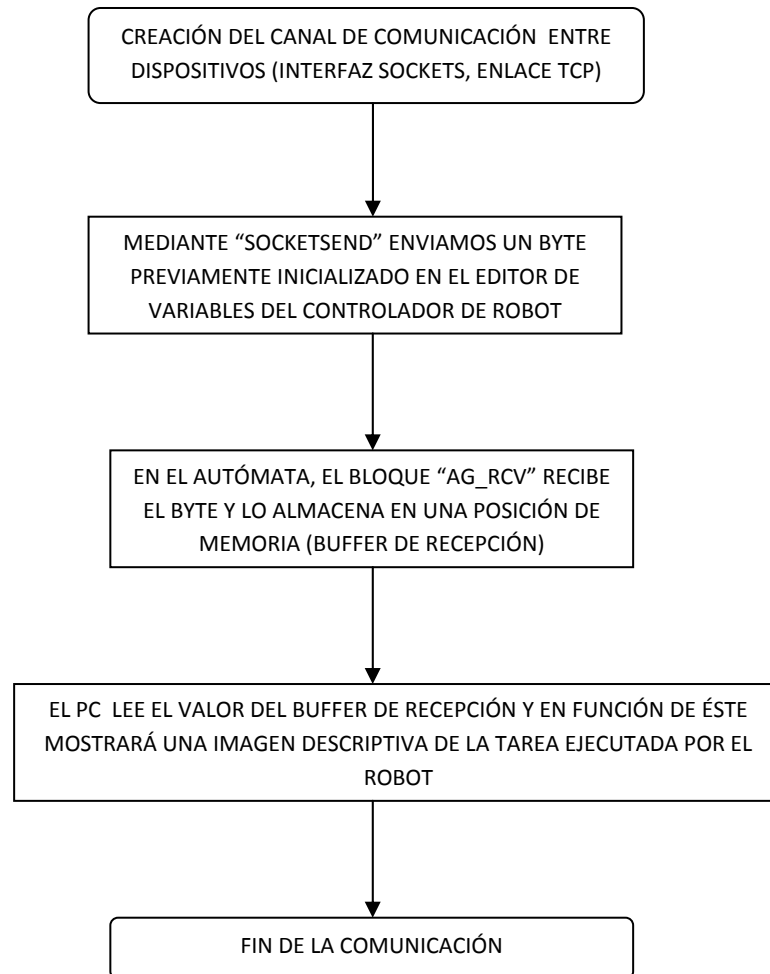


Figura 4.91. Diagrama de flujo; Envío de datos Controlador de Robot → PC.

Con los tres elementos hardware conectados por cables Ethernet mediante la interfaz *sockets* y el enlace TCP, la comunicación en el controlador de robot comienza al ejecutar la instrucción "SocketSend". El byte enviado es recibido en el autómata por medio del bloque "AG_RCV", que guardará la información en la posición de memoria del autómata denominada "Buffer de recepción". En el PC, WinCC estará leyendo constantemente dicho buffer, y en función del valor almacenado en éste, mostrará una u otra imagen representativa de la tarea ejecutada por el robot (ver tablas 4.1 y 4.2).



A continuación se muestran dos tablas resumen, la primera sobre las variables de entrada y salida utilizadas en el sistema, y la segunda con la dirección IP de cada dispositivo presente en la red Ethernet creada.

| VARIABLE | TIPO | DIRECCIÓN | FUNCIÓN |
|---------------------|----------------|----------------|---|
| ActivarComEthernet | Bit de salida | 124.1 | Bit que marca el inicio del envío de datos por Ethernet desde el PLC |
| MovImagenRobot | Byte | DB201, DBB100 | Buffer de recepción de datos del PLC. Leída desde el PC, en función de su valor se mostrará una imagen descriptiva de la tarea ejecutada por el robot |
| Estructura/Baldosas | Bits de salida | 125.1 125.2 | Orden de ensamblar estructura (desde PC) Orden de cortar baldosas (desde PC) |
| DatosRecibidosMPI | Bit de entrada | 124.0 | Indica la recepción de datos recibidos en el PLC por la interfaz MPI |

Tabla 4.1. Variables de entrada y salida.

| DIRECCIÓN IP | DISPOSITIVO DE LA RED ETHERNET |
|---------------|---|
| 192.168.133.1 | Módulo Procesador de Comunicaciones del PLC |
| 192.168.133.2 | Controlador de Robot ABB IRC5 |
| 192.168.133.3 | PC – SCADA |
| 192.168.133.5 | Módulo Switch Industrial Ethernet del PLC |

Tabla 4.2. Direcciones IP de los dispositivos de red.



4.3.3. Ejemplo de aplicación SCADA

Para probar el estado de las comunicaciones, en el PC se ha desarrollado lo que se conoce como un sistema SCADA (*Supervisory Control and Data Acquisition*), que es mostrado a continuación, tras una breve introducción sobre los sistemas SCADA.

Como se ha comentado en anteriores puntos, un PC forma parte habitualmente de los Sistemas Distribuidos de Control. No obstante su función no es la de control directo de la planta como es el caso de los autómatas programables, reguladores y demás nodos actuadores, ni la de la medición directa de las variables de planta como es el caso de los nodos sensores. Su función más habitual dentro de los sistemas distribuidos es la de ejecutar un programa SCADA, mediante el cual convertimos al PC en un puesto de monitorización y control de la instalación. Por un lado el software permite comunicar al PC con los nodos del Sistema Distribuido y de este modo el PC realiza la adquisición de datos de la planta a través de la red de comunicaciones. Por otro lado bajo control de este software en la pantalla de alta resolución del PC se representa la planta gráficamente junto con los valores de diferentes variables del proceso. Además el operador puede interactuar con esta representación gráfica de la planta a través del ratón y del teclado del PC, pudiendo tanto cambiar el estado de salidas digitales (por ejemplo ordenar abrir o cerrar una válvula) como cambiar consignas en reguladores, utilizando el PC la red para comunicar estas acciones a los nodos pertinentes. Además de las anteriores un programa SCADA tiene entre otras las siguientes funciones:

- Supervisión remota de instalaciones: mediante esta función el usuario puede conocer el estado de la planta y coordinar eficientemente las labores de producción y mantenimiento de las instalaciones.
- Control remoto de instalaciones: además del control remoto a petición del operador comentado anteriormente, los programas SCADA permiten ejecutar automáticamente acciones de mando preprogramadas (por ejemplo abrir/cerrar una válvula, arrancar/parar una bomba, etc.) dependiendo de valores actuales de las variables de la planta o de combinaciones de éstas.

También de modo automático y dependiendo de condiciones preprogramadas pueden cambiar parámetros del sistema (por ejemplo cambiar los márgenes de valores para la activación de una alarma).

- Presentación de gráficos dinámicos: las pantallas gráficas que representan la planta tienen animación. Además de actualizar periódicamente el valor de todas las variables de la planta mostradas, puede haber animación de figuras y dibujos, cambios de color, etc. (por ejemplo pueden aparecer pilotos intermitentes para indicar alarmas).
- Presentación de alarmas: se alerta al operador mediante intermitencias, cambios de color, señales acústicas, etc. de situaciones anómalas, de modo que éste pueda tomar las acciones correctoras oportunas.
- Almacenamiento de información histórica: registra y almacena datos de planta a intervalos periódicos y/o datos de alarma, etc.

No solo las empresas fabricantes de autómatas comercializan programas SCADA, sino que existen empresas especializadas en este paquete software.

La aplicación SCADA desarrollada se compone de cuatro pantallas:



Figura 4.92. Pantalla 1 del SCADA ejemplo.



En la pantalla de inicio simplemente se muestra el logotipo de ManuBuild, un proyecto sobre la aplicación de la Automatización Industrial y de la Robótica en el ámbito de la construcción.

Si pulsamos sobre “ACCEDER” pasaremos a la pantalla 2.

Si pulsamos sobre “SALIR” cerramos la aplicación.

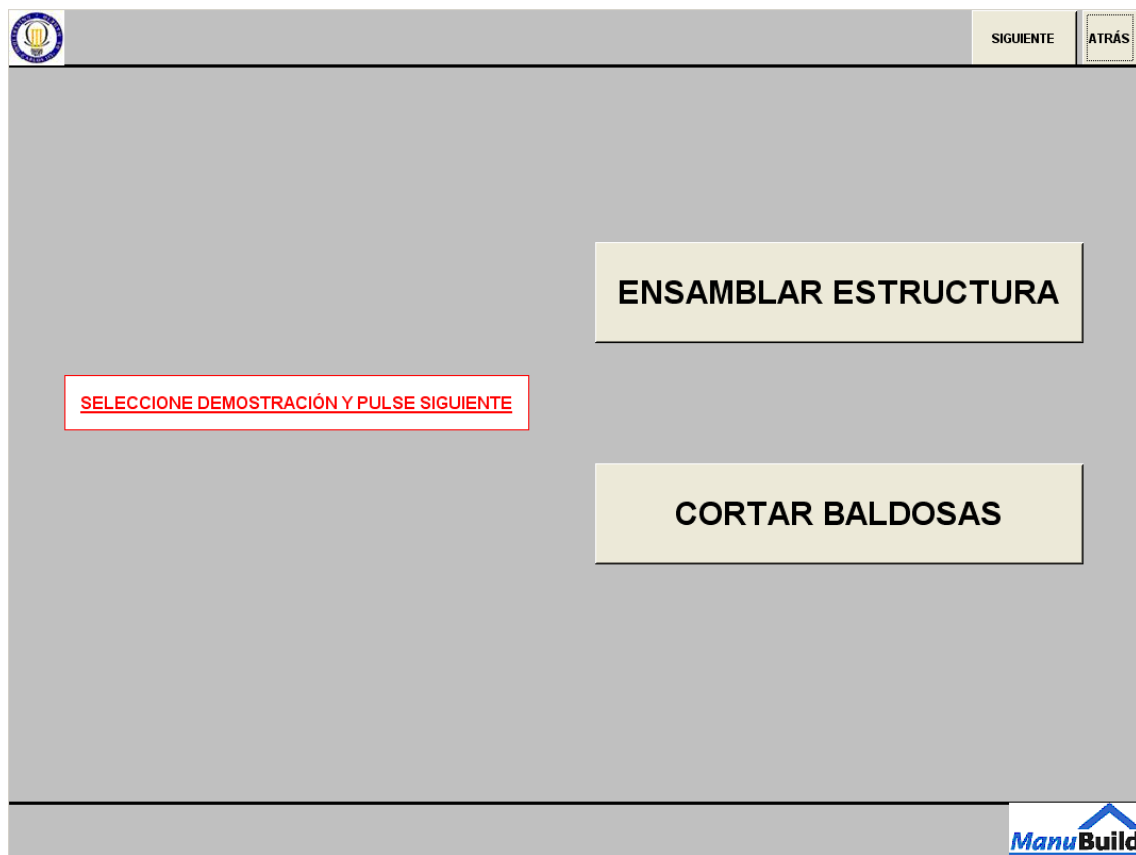


Figura 4.93. Pantalla 2 del SCADA ejemplo.

En esta pantalla seleccionaremos qué tarea queremos que realice el robot: ensamblar una estructura o cortar baldosas. Al hacer clic sobre la opción elegida, la información será transmitida al autómata, quien a su vez enviará los datos pertinentes al controlador de robot.

Si pulsamos “ATRÁS” volveremos a la pantalla 1.

Si pulsamos “SIGUIENTE” pasaremos a la pantalla 3.

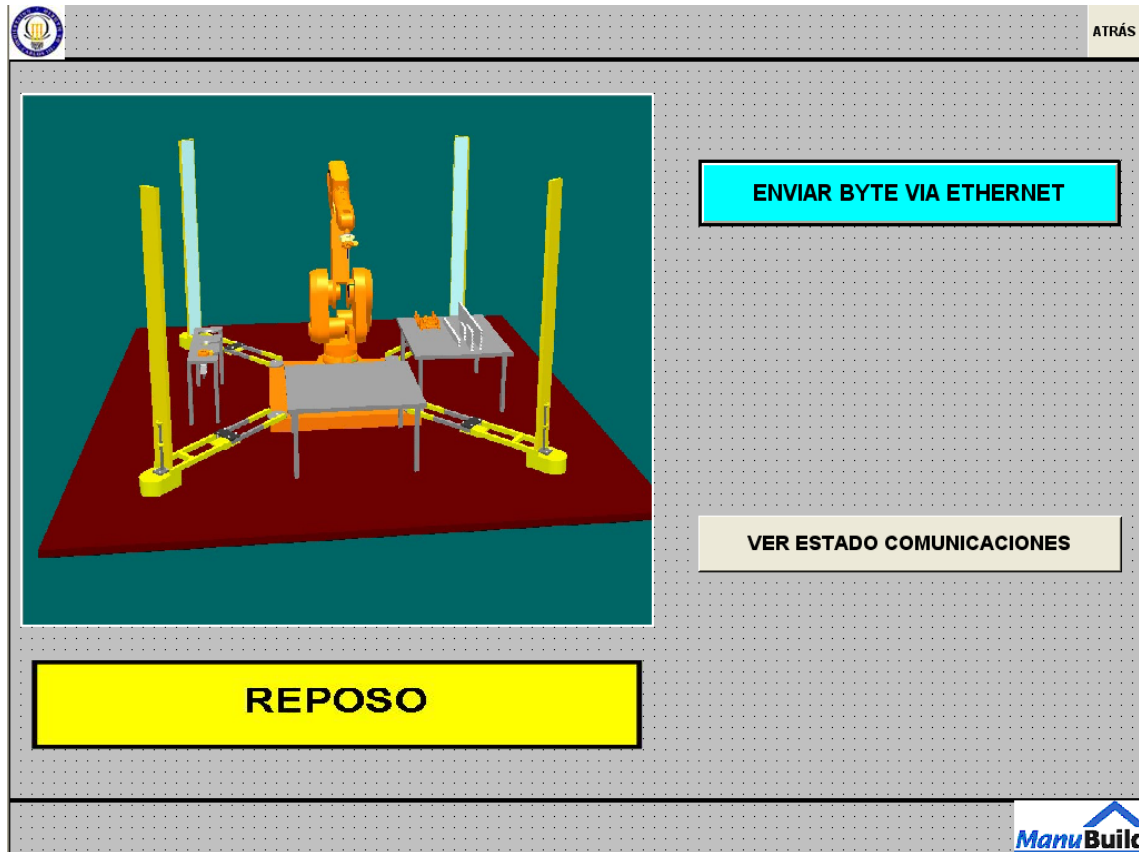


Figura 4.94. Pantalla 3 del SCADA ejemplo.

Desde esta pantalla, podemos realizar tres operaciones:

- Visualizar el estado del proceso, es decir, saber qué operación está realizando el robot en todo momento, mediante una imagen del robot y un texto descriptivo de la tarea realizada. Como es lógico, para llevar a cabo esta animación, el IRC5 deberá estar transmitiendo información en todo momento sobre qué tarea está ejecutando.
- Transmitir datos por la interfaz Industrial Ethernet, pulsando “ENVIAR BYTE VIA ETHERNET”.

- Diagnosticar si el estado de las comunicaciones (Industrial Ethernet y MPI) es correcto o no, pulsando en “VER ESTADO COMUNICACIONES”.

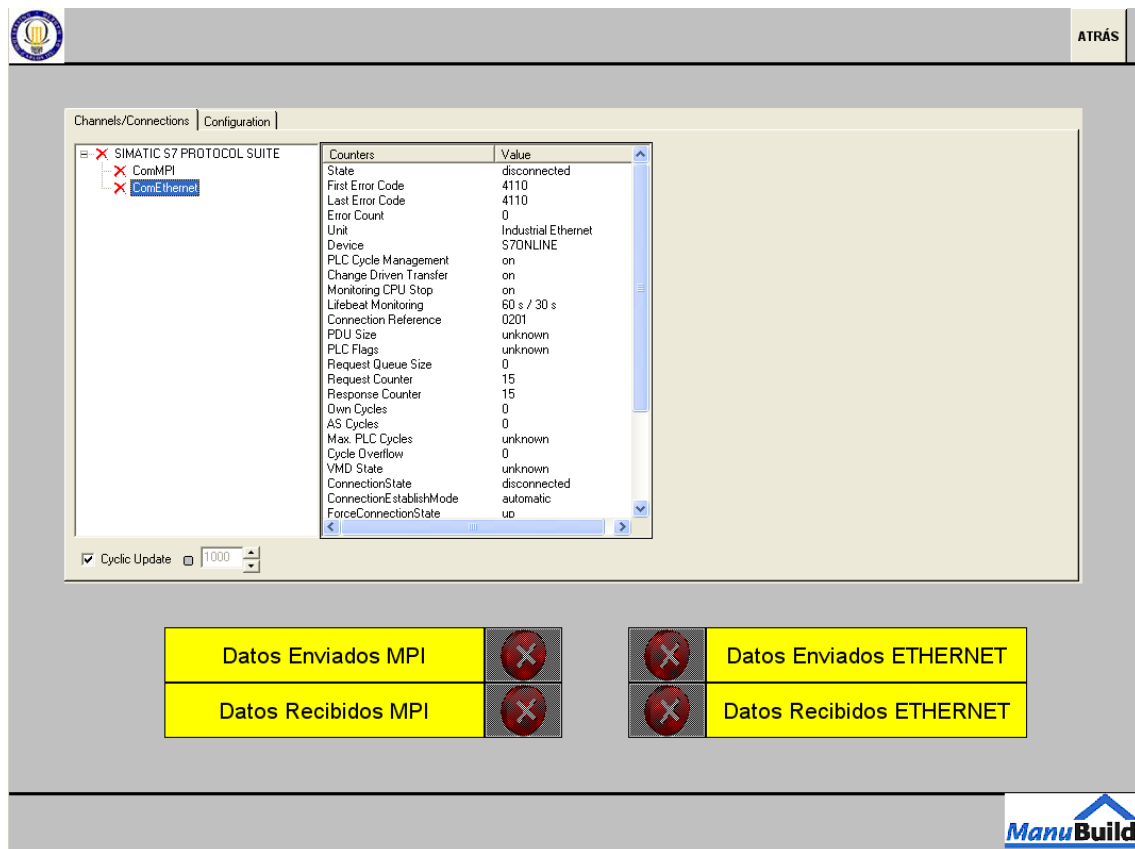


Figura 4.95. Pantalla 4 del SCADA ejemplo.

Desde esta pantalla podremos comprobar si hemos enviado o recibido datos tanto por Industrial Ethernet como por MPI. También pueden verse distintos parámetros característicos de las redes de comunicación.

En “Cyclic Update” podemos ajustar cada cuanto tiempo queremos que la aplicación SCADA se actualice.

En el sistema SCADA diseñado, para saber en qué estado se encuentra el proceso llevado a cabo por el robot, se muestra una imagen descriptiva de la tarea ejecutada por éste (la variable empleada en WinCC para este fin es de tipo byte: *MovImagenRobot*). Dicho trabajo consiste en el ensamblaje de una estructura. La secuencia de imágenes es la siguiente:



| | | | | |
|--|--|--|---|--|
| ESTADO 0 | ESTADO 1 | ESTADO 2 | ESTADO 3 | ESTADO 4 |
|  MovImagenRobot=0 |  MovImagenRobot=1 |  MovImagenRobot=2 |  MovImagenRobot=3 |  MovImagenRobot=4 |
| ESTADO 5 | ESTADO 6 | ESTADO 7 | ESTADO 8 | ESTADO 9 |
|  MovImagenRobot=5 |  MovImagenRobot=6 |  MovImagenRobot=7 |  MovImagenRobot=8 |  MovImagenRobot=9 |
| ESTADO 10 | ESTADO 11 | ESTADO 12 | ESTADO 13 | ESTADO 14 |
|  MovImagenRobot=10 |  MovImagenRobot=11 |  MovImagenRobot=12 |  MovImagenRobot=13 |  MovImagenRobot=14 |
| ESTADO 15 | ESTADO 16 | ESTADO 17 | ESTADO 18 | ESTADO 19 |
|  MovImagenRobot=15 |  MovImagenRobot=16 |  MovImagenRobot=17 |  MovImagenRobot=18 |  MovImagenRobot=19 |
| ESTADO 20 | ESTADO 21 | ESTADO 22 | ESTADO 23 | ESTADO 24 |
|  MovImagenRobot=20 |  MovImagenRobot=21 |  MovImagenRobot=22 |  MovImagenRobot=23 |  MovImagenRobot=24 |

Tabla 4.1. Vistas de estado del proceso 1/2.



| ESTADO 25 | ESTADO 26 | ESTADO 27 | ESTADO 28 | ESTADO 29 |
|--|--|--|---|--|
|  MovImagenRobot=25 |  MovImagenRobot=26 |  MovImagenRobot=27 |  MovImagenRobot=28 |  MovImagenRobot=29 |
| ESTADO 30 | ESTADO 31 | ESTADO 32 | ESTADO 33 | ESTADO 34 |
|  MovImagenRobot=30 |  MovImagenRobot=31 |  MovImagenRobot=32 |  MovImagenRobot=33 |  MovImagenRobot=34 |
| ESTADO 35 | ESTADO 36 | ESTADO 37 | ESTADO 38 | ESTADO 39 |
|  MovImagenRobot=35 |  MovImagenRobot=36 |  MovImagenRobot=37 |  MovImagenRobot=38 |  MovImagenRobot=39 |
| ESTADO 40 | ESTADO 41 | ESTADO 42 | ESTADO 43 | ESTADO 44 |
|  MovImagenRobot=40 |  MovImagenRobot=41 |  MovImagenRobot=42 |  MovImagenRobot=43 |  MovImagenRobot=44 |
| ESTADO 45 | ESTADO 46 | ESTADO 47 | ESTADO 48 | |
|  MovImagenRobot=45 |  MovImagenRobot=46 |  MovImagenRobot=47 |  MovImagenRobot=48 | |

Tabla 4.2. Vistas de estado del proceso 2/2.



Capítulo 5:

Conclusiones.

En primer lugar hay que decir que se han cumplido los objetivos inicialmente propuestos, es decir, se ha logrado establecer una comunicación básica, mediante la interfaz Ethernet Industrial, entre los tres elementos que componen nuestra red: PC, autómatas y controlador de robot IRC5.

Hemos conseguido una comunicación full dúplex. Podemos enviar un byte desde el PC y recibirlo en el controlador de robot, pasando previamente por el autómata. Y del mismo modo, podemos enviar un byte desde el controlador de robot y leer su valor desde el PC.

Todo este trabajo fue desarrollado con algunas dificultades y problemas que a continuación se exponen, aportándose también la solución a los mismos:

- Para configurar la interfaz de comunicación del PC con el autómata en el software Simatic Manager (Herramientas→Ajustar interfaz PG/PC) en un primer momento no era posible establecer la tarjeta de red Ethernet del PC como interfaz, debido a la versión del software. Este hecho fue solucionado instalando la **versión 5.4 incl. SP3 del Simatic Manager**.
- Un problema que nos supuso bastante tiempo de trabajo, fue el hecho de cómo agregar el ordenador a la red Ethernet creada, debido a que en el software SIMATIC Manager existían distintos objetos y configuraciones posibles para representar un PC en la red Ethernet. Esto es posible mediante el objeto PG/PC disponible en el Netpro, en el que fijaremos la dirección IP que leeremos en la conexión de nuestro PC, y que deberá estar en concordancia con el resto de elementos de la red.
- En la recepción de datos por parte del autómata, el buffer configurado era mayor que un byte, que es lo que envía el robot, y este hecho ocasionaba que hasta que el buffer de recepción no se completaba en su totalidad, los datos recibidos no eran visibles. La cuestión clave es, por tanto, enviar desde el robot la misma cantidad de bytes que espera el autómata, y viceversa.



Por último se hablará sobre las conclusiones obtenidas del empleo de la tecnología Ethernet como red de comunicación industrial.

Ethernet Industrial es ya una tecnología potente en el entorno industrial. El hecho de que sea fácilmente instalable y configurable ha hecho que numerosos fabricantes desarrollen la mayoría de sus productos con este tipo de tecnología.

La facilidad de administración, consulta o programación de los dispositivos aporta a la empresa muchas ventajas y ahorros económicos. El hecho de tener acceso a la red industrial mediante un navegador Web facilita el trabajo de los operadores. El uso de la infraestructura Ethernet facilita también los tediosos diagnósticos del medio físico y el cambio de conectores, cableado... que suelen ser el 80% de las veces la causa de errores en los buses de campo.

Ethernet industrial es una realidad hoy, y gracias a todos los beneficios y ventajas que ofrece será estudiada por las empresas, de manera que cada vez sea más útil en el sector industrial llegando a construirse todo tipo de dispositivos para esta tecnología.



Capítulo 6:
Trabajos futuros.



En este apartado se hablará sobre los posibles trabajos futuros que podrían llevarse a cabo sobre la base de lo desarrollado en este proyecto:

- ✓ Ampliar la red con nuevos equipos como PCs, robots o autómatas, de tal forma que se aumentaría el número de dispositivos controlados desde un mismo punto central.
- ✓ Para mejorar la seguridad de las comunicaciones, podrían implementarse mecanismos de redundancia con el fin de que el factor ruido no provoque pérdidas de información.
- ✓ El sistema SCADA desarrollado podría mejorarse añadiendo nuevas funcionalidades como:
 - Saber qué herramienta está utilizando el robot (pinza, ventosas, taladro, etc.).
 - Parámetros de la herramienta utilizada (velocidad del motor, temperatura, etc.).
 - Mostrar la instrucción de código que se está ejecutando.
 - Gráficas y tablas de valores medidos.
 - Configurar permisos y accesos de los distintos usuarios de la aplicación.
 - Mostrar gráficas con históricos, tablas con alarmas y eventos, tendencias, mensajes, etc.

Sobre la base del trabajo desarrollado, estas mejoras del SCADA podrían implementarse de un modo relativamente sencillo. En este proyecto se proporciona toda la información básica necesaria para crear en el futuro un sistema SCADA más perfeccionado para el proyecto Manubuild.



Capítulo 7:

Bibliografía.



1. Comunicaciones industriales: principios básicos.

Castro Gil, Manuel-Alonso.

2. Comunicaciones industriales: sistemas distribuidos y aplicaciones.

Castro Gil, Manuel-Alonso.

3. Comunicaciones industriales con SIMATIC S7.

Víctor Sempere Paya, Sergio Cerdá Fernández.

Escuela Politécnica Superior de Alcoy.

4. Apuntes de Sistemas Informáticos en Tiempo Real.

Universidad Carlos III de Madrid.

Departamento de Ingeniería de Sistemas y Automática.

5. CD Documentación del Robot ABB IRB 2400.

6. Programa Step7 del autómeta:

<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&objId=17853532&objAction=csOpen&lang=es&siteid=cseus&aktprim=0&extranet=standard&viewreg=WW>

7. Ethernet industrial:

7.1. Ethernet Industrial de Siemens:

http://www.automation.siemens.com/net/html_78/produkte/040_produkte.htm

7.2. Switch Ethernet Industrial SCALANCE X208:

http://www.profibus.com/member/siemens_ag/products/article/5944/



7.3. Protocolos de comunicación soportados:

<https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&objId=16767769&objAction=csOpen&nodeid0=10805161&lang=en&siteid=cseus&aktprim=0&extranet=standard&viewreg=WW>

7.4. Configuración del enlace TCP con un PC con interfaz *sockets*:

<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=22790099&caller=view>

7.5. Ajustar la interfaz Ethernet del PC:

http://www.lawebdelprogramador.com/news/mostrar_new.php?id=204&texto=SCADA&n1=277935&n2=0&n3=0&n4=0&n5=0&n6=0&n7=0&n8=0&n9=0&n0=0

7.6. Tecnología Ethernet:

<http://es.wikipedia.org/wiki/Ethernet>

7.7. Cables RJ-45 directos y cruzados:

<http://es.wikipedia.org/wiki/RJ-45>

7.8. Hardware asociado a una red Ethernet:

<http://es.wikipedia.org/wiki/Switch>

8. Conceptos sobre “Comunicaciones Industriales”:

8.1. Sistemas de Comunicaciones Industriales:

tec.upc.es/ie/practi/Sistemas.pdf



- 8.2. Instalación de Sistemas de Automatización y Datos, Universidad de Vigo:

http://www.uvigo.tv/uploads/material/Video/1567/ISAD_Tema6.pdf

- 8.3. Introducción a las redes de comunicación industriales:

http://www.disa.bi.ehu.es/spanish/ftp/material_asignaturas/Laboratorio%20de%20Comunicaciones%20Industriales/Documentaci%F3n/Introducci%F3n%20a%20las%20Comunicaciones%20Industriales.pdf

- 8.4. Buses de campo:

<http://linux0.unsl.edu.ar/~rvilla/c3m10/tema13.pdf>

- 8.5. Introducción a los sistemas de tiempo real:

<http://laurel.datsi.fi.upm.es/~ssoo/STR/Introduccion.pdf>

- 8.6. Modelo OSI:

http://es.wikipedia.org/wiki/Modelo_OSI

- 8.7. Red industrial:

http://es.wikipedia.org/wiki/Red_industrial

9. WinCC:

- 9.1. Manual de uso e iniciación:

http://formacion.plcmadrid.es/descargas/docs/manuales/SID_practica7_WinCC_iniciacion.pdf

- 9.2. Programación Elemental:

http://www.disa.bi.ehu.es/spanish/ftp/material_asignaturas/Tecnolog%EDas%20de%20Control/Laboratorio/WinCC.pdf



10. <http://www.abb.es/product/seitp327/f597ea096948f833c125717900224b21.aspx?productLanguage=es&country=ES> (**ROBOT ABB IRB 2400**)
11. <http://www.abb.es/product/seitp327/c856731f190645eec125716c0023a375.aspx?productLanguage=es&country=ES> (**CONTROLADOR DE ROBOT IRC5**)
12. http://www.infoplcn.net/Enlace/DOC_ROBOTICA/ABB/infoPLC_net_Introduccion_RAPID.html
13. <http://es.wikipedia.org/wiki/TCP>
14. <http://es.wikipedia.org/wiki/WAN>
15. http://es.wikipedia.org/wiki/Socket_de_Internet
16. http://es.wikipedia.org/wiki/Caracteristicas_p2p
17. http://es.wikipedia.org/wiki/Windows_CE
18. http://es.wikipedia.org/wiki/Familia_de_protocolos_de_Internet#Ventajas_e_inconvenientes
19. [http://es.wikipedia.org/wiki/D%C3%BAplex_\(telecomunicaciones\)](http://es.wikipedia.org/wiki/D%C3%BAplex_(telecomunicaciones))
20. <http://www.casadomo.com/noticiasDetalle.aspx?c=143&idm=152&m=164&n2=148&pat=148>
21. <http://es.wikipedia.org/wiki/Automatizaci3n>
22. http://es.wikipedia.org/wiki/Controlador_l3gico_programable
23. <http://es.wikipedia.org/wiki/SCADA>
24. <http://www.manubuild.org/summary>